Article

Hidden Markov Modeling with Detailed Balance and Its Application to Single Protein Folding

Yongli Zhang,[1,*] Junyi Jiao,[1,2] and Aleksander A. Rebane[1,2,3,4]

[1]Department of Cell Biology, School of Medicine, [2]Integrated Graduate Program in Physical and Engineering Biology, and [3]Department of Physics, Yale University, New Haven, Connecticut; and [4]Nanobiology Institute, Yale University, West Haven, Connecticut

ABSTRACT   Hidden Markov modeling (HMM) has revolutionized kinetic studies of macromolecules. However, results from HMM often violate detailed balance when applied to the transitions under thermodynamic equilibrium, and the consequence of such violation has not been well understood. Here, to our knowledge, we developed a new HMM method that satisfies detailed balance (HMM-DB) and optimizes model parameters by gradient search. We used free energy of stable and transition states as independent fitting parameters and considered both normal and skew normal distributions of the measurement noise. We validated our method by analyzing simulated extension trajectories that mimicked experimental data of single protein folding from optical tweezers. We then applied HMM-DB to elucidate kinetics of regulated SNARE zippering containing degenerate states. For both simulated and measured trajectories, we found that HMM-DB significantly reduced overfitting of short trajectories compared to the standard HMM based on an expectation-maximization algorithm, leading to more accurate and reliable model fitting by HMM-DB. We revealed how HMM-DB could be conveniently used to derive a simplified energy landscape of protein folding. Finally, we extended HMM-DB to correct the baseline drift in single-molecule trajectories. Together, we demonstrated an efficient, versatile, and reliable method of HMM for kinetics studies of macromolecules under thermodynamic equilibrium.

INTRODUCTION

Hidden Markov modeling (HMM) and maximum-likelihood estimation have been widely used in bioinformatics, image processing, speech recognition, prediction of election results, and other disciplines (1–3). In particular, HMM becomes increasingly important for analyses of single-molecule trajectories measured by patch-clamps (4,5), single-molecule fluorescence resonance energy transfer or tracking (6–9), single-molecule force spectroscopy (10–14), etc. Compared with traditional experimental approaches based on an ensemble of molecules, single-molecule methods can reveal more intermediate states and more detailed transition kinetics in real time. In addition, single-molecule methods can detect kinetics of macromolecular transitions (such as protein folding) under equilibrium conditions, and reveal both energetics and kinetics of the associated transitions from a single experiment (15,16). As the resolution of single-molecule measurements increases, complex reaction networks involving many states and multiple timescales have been studied. For example, latest high-resolution optical tweezers (OTs) are capable of

detecting distance changes ranging from a few Ångstroms to tens of microns in a timescale from a few microseconds to hours (17–19). As a result, equilibrium protein folding and unfolding has been observed among six states that significantly differ in their energies and lifetimes (11). Dissecting such reaction networks from the single-molecule trajectories imposes a challenge for data analysis, and requires improved HMM.

A hidden Markov model with detailed balance is crucial to modeling transitions at equilibrium. Detailed balance is both a necessary and a sufficient condition for any system in thermal equilibrium (20,21). Detailed balance requires the probability flux from state i to state j to be equal to the flux in a reverse direction, or $p_i q_{ij} = p_j q_{ji}$, for any pair of states, where p_i is the probability being in state i, and q_{ij} is the rate constant of the transition from state i to state j. As a result, any net probability flux in the reaction network, including steady-state flux along any closed loop, is prohibited. In contrast, the standard maximum likelihood algorithms used in HMM only ensure that the system be at steady state (1,4,22). For any linear models corresponding to sequential protein folding and unfolding, a steady state is equivalent to an equilibrium state. Consequently, the HMM results of these models automatically satisfy detailed balance. However, in more complex

---

CrossMark

Markov models at steady state, flux can still occur along closed loops in the reaction network. Therefore, detailed balance can impose strong constraints on a Markov model. For example, for a seven-state Markov model, detailed balance adds a maximum of 15 constraints to the model parameters.

Incorporating detailed balance into HMM remains a challenge. HMM models a hidden kinetic process by a Markov model and its associated observable obscured by the measurement noise by a state-dependent statistical model (1,6,8). In protein folding studied by optical tweezers, the measurement or observable is typically the total extension of a single protein and the linker required to attach the protein to two beads (23,24). The protein folding process becomes hidden also because two or more protein folding states may exhibit identical average extensions (25–27). Both the measurement noise and state degeneracy make it difficult to determine protein folding kinetics from the measured extension. Given a single-molecule trajectory and a hidden Markov model, the likelihood of observing the trajectory can be efficiently calculated by the forward-backward algorithm (1). Then the likelihood can be maximized to reveal model parameters that best fit the measured trajectory, a process called maximum-likelihood estimation. The maximization is typically implemented either by an expectation-maximization (EM) algorithm or by a gradient search method (4,8,22). The EM algorithm directly optimizes the state transition probabilities, and detailed balance constrains the transition probabilities by the highly nonlinear Wegscheider's equations (20). For instance, a three-state Markov model has a single constraint $q_{12}q_{23}q_{31} = q_{13}q_{32}q_{21}$. To our knowledge, no detailed balance has been directly incorporated into the EM algorithm. In contrast, the gradient search method permits various constraints. Qin et al. (4) demonstrated that the gradient search method offers several advantages over the EM algorithm, including fast convergence and high accuracy for trajectories that have a small signal/noise ratio or a low sampling rate. They also proposed a scheme to implement detailed balance in the gradient search method. Nevertheless, the effect of detailed balance in HMM has not been demonstrated, especially when the number of states becomes large (20). A Bayesian extension of HMM based on Gibbs sampling has been reported (12), which potentially allows incorporation of detailed balance. Again, the specific effect of detailed balance has not been examined.

Motivated by the work of Qin et al. (4), we developed, to our knowledge, a new HMM with detailed balance (HMM-DB) using energy of stable states and transition states as independent model parameters as well as fitting parameters. As a result, our method did not require constraints from detailed balance and could be easily extended to HMM of large reaction networks. We demonstrated the advantages of HMM-DB by analyzing simulated and measured single-molecule trajectories from OTs, considering both normal and skew normal noise distributions. We found that HMM-DB yields more reliable results than HMM-EM without detailed balance.

MATERIALS AND METHODS

HMM for single protein folding detected by OTs

Suppose a protein undergoes reversible folding and unfolding transitions among N distinct states under approximately constant force and that the transitions are described by the rate constant matrix $\mathbf{q} = [q_{ij}]_{N \times N}$, where q_{ij} denotes the rate constant for the transition from state i to state j. The transition probability $\mathbf{a}$ among these states over a data sampling time Δt can be solved from the master equation $d\mathbf{a}/dt = \mathbf{aq}$, or

$$\mathbf{a} = \exp(\mathbf{q}\Delta t). \tag{1}$$

Here, $\mathbf{a} = [a_{ij}]_{N \times N}$ is the transition probability matrix, whose element a_{ij} represents the probability of the protein being in state j at time Δt, given that the protein occupies state i at time zero. The matrix $\mathbf{a}$ in Eq. 1 can be conveniently computed using the matrix exponential function expm or eigenvalue decomposition as shown in the Supporting Materials and Methods. Under the condition $\mathbf{q}\Delta t \ll 1$,

$$\mathbf{a} \approx \mathbf{I} + \mathbf{q}\Delta t, \tag{2}$$

where $\mathbf{I}$ is the identity matrix. HMM-EM estimates the best-fit matrix $\mathbf{a}$ from a time series. To calculate the rate constant matrix $\mathbf{q}$ from $\mathbf{a}$ using Eq. 1, we used the software MATLAB's matrix logarithm function log$m$, or $\mathbf{q} = \log m(\mathbf{a})/\Delta t$ (The MathWorks, Natick, MA). However, the rate constants corresponding to small transition probability elements in $\mathbf{a}$ calculated by this method tended to be negative. In this case, we used Eq. 2 to calculate an approximate transition rate constant matrix $\mathbf{q}$.

OTs record the time-dependent extension of a single protein during its reversible folding and unfolding among at least two states. In each state, the measured extension fluctuates around a mean extension. To a first-order approximation, the fluctuation can be represented by a normal distribution (10,28,29). Suppose the protein in state i has an average extension μ_i, with fluctuation σ_i; then the probability to observe an instantaneous extension x can be expressed as

$$b_i(x) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{(x - \mu_i)^2}{2\sigma_i^2} \right]. \tag{3}$$

The extension fluctuation can be approximated by $\sigma_i^2 = k_B T/k_i$ with $k_i = k_{tether} + k_{trap}$, where k_{tether} and k_{trap} are the effective force constants of the protein-DNA tether and two optical traps, respectively. The value k_{trap} is related to the force constants of the two optical traps k_a and k_b by $k_{trap} = k_a k_b/(k_a + k_b)$, which is state independent. However, k_{tether} generally decreases as force decreases or protein unfolds. Thus, the extension fluctuation σ_i is state dependent. The probability density function (PDF) $\rho(x)$ of the extension with multiple states can be expressed as

$$\rho(x) = \sum_{i=1}^{N} p_i b_i(x). \tag{4}$$

This PDF can be compared to the PDF of an extension trajectory, which is calculated as the number of extension data points N_i falling into the ith bin with an average extension of x_i, i.e., $\rho_i = N_i/(N_T dx)$, where N_T is the total number of data points in all bins and dx is the extension size for each bin.

HMM with a skew noise distribution

We adopted the skew normal distribution to describe the skew extension fluctuation (30). The PDF is expressed as

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \left\{ 1 + \mathrm{erf}\left[\frac{\zeta(x-\mu)}{\sqrt{2}\sigma} \right] \right\}, \quad (5)$$

where the shape parameter ζ characterizes the skewness of the distribution and

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (6)$$

is the error function, which is a built-in function in MATLAB. The mean $\bar{x}$ and the standard deviation ω of the extension x are

$$\bar{x} = \mu + \sigma\delta\sqrt{\frac{2}{\pi}} \quad (7)$$

and

$$\omega \equiv \left[\overline{(x-\bar{x})^2} \right]^{1/2} = \sigma\sqrt{1 - \frac{2\delta^2}{\pi}}, \quad (8)$$

respectively, with $\delta = \zeta/\sqrt{1+\zeta^2}$. In addition, the skewness γ of the PDF can be calculated by

$$\gamma = \overline{\left(\frac{x-\bar{x}}{\omega} \right)^3} = \frac{4-\pi}{2} \frac{\left(\sqrt{\frac{2}{\pi}}\delta \right)^3}{\left(1 - \frac{2\delta^2}{\pi} \right)^{3/2}}. \quad (9)$$

Therefore, the extension of each protein state is characterized by the three parameters (μ, σ, and ζ). A derivation of Eq. 7 is shown in the Supporting Materials and Methods.

Markov model defined on the simplified energy landscape

Given the energy matrix **g** for an N-state hidden-Markov model, the equilibrium state population $\mathbf{p} = [p_i]_{N \times 1}$ is determined by the Boltzmann distribution, i.e.,

$$\mathbf{p} = \frac{1}{\sum_{i=1}^{N} e^{-\frac{g_{ii}}{k_B T}}} \begin{pmatrix} e^{-\frac{g_{11}}{k_B T}} \\ e^{-\frac{g_{22}}{k_B T}} \\ \vdots \\ e^{-\frac{g_{NN}}{k_B T}} \end{pmatrix}. \quad (10)$$

Here k_B is the Boltzmann constant, and T is the temperature. To derive the transition rates from the energy matrix **g**, we started from the Kramers rate equation

$$q_{ij} = \frac{D\sqrt{\kappa_i \kappa_{ij}}}{2\pi k_B T} \exp\left(-\frac{\Delta G_{ij}^{\ddagger}}{k_B T} \right), \quad (11)$$

where D is the diffusion constant over an energy barrier; κ and κ_{ij} are the curvatures of the continuous free energy landscape at the energy minimum G_i and the energy barrier G_{ij} corresponding to the stable state i (Fig. 1 B, point A) and the transition state from state i to state j (point B), respectively; and $\Delta G_{ij}^{\ddagger} = G_{ij} - G_i$ is the corresponding activation energy. The minima and maxima of the energy landscape are often interpreted as the free energy of the stable states and the transition states, respectively. However, based on this interpretation, the Kramers' equation is not compatible with detailed balance, because the net probability flux from state i to state j is

$$p_i q_{ij} - p_j q_{ji} \propto \frac{D\sqrt{\kappa_{ij}}}{2\pi k_B T} \left(\sqrt{\kappa_i} - \sqrt{\kappa_j} \right) \exp\left(-\frac{G_{ij}}{k_B T} \right) \neq 0. \quad (12)$$

The inequality in Eq. 12 is generally true, because the curvatures corresponding to different stable states are typically not equal. For example, proteins in folded states tend to be more rigid than in their unfolded states, leading to greater curvatures of the potential wells for the folded states than for the unfolded states.

To resolve the above contradiction, we noticed that a discrete stable state is not a single state represented by the minimum of an energy well in the continuous energy landscape $G(x)$, but contains all states in the energy well. Thus, the free energy of the stable i, g_i, is not equal to G_i (Fig. 1 B). Instead, based on the Boltzmann distribution,

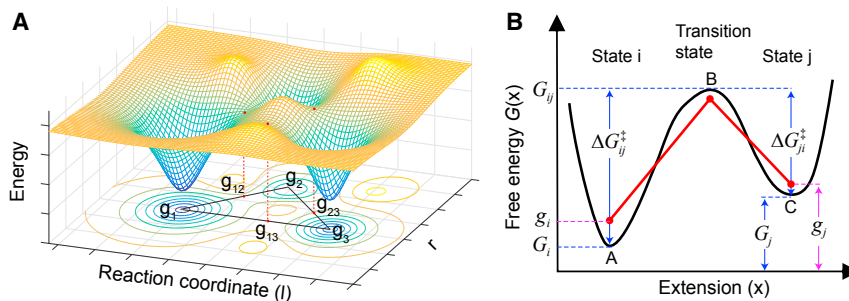$$g_i \propto -k_B T \ln \int_{\text{well}} \exp\left[-\frac{G(x)}{k_B T} \right] dx, \quad (13)$$

FIGURE 1 HMM and energy landscape of protein folding. (A) A schematic energy landscape for three-state protein folding is shown as a function of two orthogonal reaction coordinates l and r. The energy landscape is characterized by wells and saddle points (marked by *red points*) representing stable states and transition states, respectively. Their corresponding discrete free energies (g_1, g_2, g_3, g_{12}, g_{23}, and g_{13}) are chosen as independent parameters for HMM with detailed balance. The reaction coordinates along the pulling direction l corresponding to all states can be determined from the measured force-dependent energy. (B) Comparison of the continuous energy landscape (G, *black curve*) and the discrete simplified energy landscape (g, *red circles*) of protein folding as a function of the protein extension. To see this figure in color, go online.

where the integral is proportional to the probability of the protein in state i and the integration is over the region corresponding to the potential well associated with state i. Approximating the energy well with a harmonic function, we derived the free energy of state i as

$$g_i = G_i + \frac{1}{2}k_B T \ln \kappa_i + C, \tag{14}$$

where C is a state-independent constant. Thus, the free energy of a stable state increases as the curvature of the potential well increases, because a narrow potential well decreases the entropy of the system that is calculated as $S_i = \frac{1}{2}k_B(1 - \ln \kappa_i) - \frac{C}{T}$.

Equation 14 reveals that the protein folding or unfolding energy defined as

$$\Delta g_{ij} = g_j - g_i = \Delta G_{ij} - \frac{1}{2}k_B T \ln \frac{\kappa_j}{\kappa_j} \tag{15}$$

generally differs from the energy difference ($\Delta G_{ij} = G_j - G_i$) derived from the Kramers equation or its variant such as the Dudko-Hummer-Szabo formula (31). Using Eq. 14, we revised the Kramers equation in Eq. 11 in the form

$$q_{ij} = k_{ij}^{(0)} \exp\left(-\frac{g_{ij} - g_{ii}}{k_B T}\right), \tag{16}$$

where $k_{ij}^{(0)}$ is the rate constant in the absence of any energy barrier and is related to the transition state energy g_{ij} by

$$k_{ij}^{(0)} \exp\left(-\frac{g_{ij}}{k_B T}\right) = \frac{D\sqrt{\kappa_{ij}}}{2\pi k_B T}\exp\left(-\frac{G_{ij}}{k_B T}\right). \tag{17}$$

Using Eqs. 10 and 16, one can show that detailed balance is now satisfied on the simplified energy landscape characterized by the energy matrix $\boldsymbol{g}$, given that $\kappa_{ij} = \kappa_{ji}$, $G_{ij} = G_{ji}$, $g_{ij} = g_{ji}$, $k_{ij}^{(0)} = k_{ji}^{(0)}$ for all $i \neq j$, $i, j = 1, 2, \cdots, N$. In general, $k_{ij}^{(0)}$ varies among transition states. In addition, an increase in $k_{ij}^{(0)}$ will increase the energy of its associated transition state g_{ij}, but not the energy of all stable states and other transition states, as is indicated by Eq. 17. As a result, different choices of $k_{ij}^{(0)}$ values affect the heights of the energy barriers in the simplified energy landscape, but not the transition rates derived from our HMM-DB (Table S6 in the Supporting Material). For simplicity, in our following analysis we assumed $k_{ij}^{(0)}$ to be the same for all transition states, i.e., $k_{ij}^{(0)} = k_m$, $i \neq j$, $i, j = 1, \cdots, N$, in which k_m represents the diffusion-limited molecular transition rate in the absence of any energy barrier. In particular, we chose $k_m = 10^6 \text{ s}^{-1}$ for the SNARE folding study described in this work as in our previous work (24,26,32,33). However, we noticed that different values for the molecular transition rate have been used or derived for folding of different proteins (11,17). In the following, we chose the unit of the energy to be the thermal energy $k_B T$. Thus, we rewrote the rate constant matrix as

$$q_{ij} = \begin{cases} k_m e^{g_{ii} - g_{ij}}, & i \neq j, \\ -k_m e^{g_{ii}} \sum_{k \neq i}^{N} e^{-g_{ik}}, & i = j, \end{cases} \quad i, j = 1, \cdots, N, \tag{18}$$

where the diagonal element of the rate constant matrix $\boldsymbol{q}$ was chosen such that each row of the matrix sums to zero (see Table 2), as is required by the master equation. Thus, our HMM-DB is defined by parameters g_{ij}, μ_i, and σ_i with $j \geq i$, $i, j = 1, \cdots, N$ for normal noise distributions and additional shape parameters ζ_i for skew normal noise distributions.

Log likelihood and its gradient

Please see the Supporting Material for their derivations and computations.

Computation of HMM-DB and HMM-EM

All computations were carried out using the desktop computer Optiplex 790 (Dell, Round Rock, TX) that contained an Intel Core i5-2500 at 3.3 GHz and 8 GB memory (Intel, Santa Clara, CA). For both HMM-DB and HMM-EM, the log-likelihood was computed using the forward-backward algorithm (Supporting Materials and Methods) and the extension idealization was calculated using the Viterbi algorithm (1). Maximum-likelihood estimation in HMM-DB was performed by gradient search using the derived gradients shown in the Supporting Materials and Methods and the MATLAB functions fminunc or fmincon. The latter function permits linear constraints, which was used to constrain equal-state extensions for the two degenerate states 1 and 2 (see Fig. 6 B). For HMM-EM, the estimation was conducted by the Baum-Welch algorithm (22). Relevant MATLAB codes will be deposited in the MATLAB Central database (http://www.mathworks.com/matlabcentral/) or will be available upon request.

Comparison between HMM-DB and HMM-EM

Because HMM-EM uses the transition probabilities as fitting parameters, we needed to covert these probabilities to the energies of stable and transition states. To this end, we first calculated the steady-state population $\mathbf{p}$ as the eigenvector of a^T, i.e., $\mathbf{a}^T \mathbf{p} = \mathbf{p}$. Then the transition rate constants were determined using Eq. 2. Finally, based on Eqs. 10 and 18, we computed the energies using the following equations:

$$g_{ij} = \begin{cases} -\ln\left(\frac{p_i}{p_N}\right), & i = j, \\ g_{ii} - \ln\frac{q_{ij}}{k_m}, & i \neq j. \end{cases} \tag{19}$$

If the system does not satisfy detailed balance, g_{ij} generally is not equal to g_{ji}. Defining

$$\overline{g}_{ij} \equiv \frac{1}{2}\left(g_{ij} + g_{ji}\right), \tag{20}$$

$$\Delta_{ij} \equiv g_{ij} - g_{ji}, \tag{21}$$

and

$$A_{ij} \equiv p_i q_{ij}, \tag{22}$$

we could quantify the deviation from detailed balance, or degree of imbalance d_{ij}, by the relative difference between the forward and backward transition rates A_{ij} and A_{ji}, i.e.,

$$d_{ij} \equiv \frac{A_{ji} - A_{ij}}{(A_{ji} + A_{ij})/2} = 2\tanh\left(\frac{1}{2}\Delta_{ij}\right) \approx \Delta_{ij}, \tag{23}$$

where the approximation is generally valid in our case, because Δ_{ij} is typically small.

SNARE zippering and SNAP binding

The SNARE constructs, experimental setup and procedures, and preliminary HMM of the extension trajectory at 1 kHz were recently described

in detail elsewhere (26). Briefly, the cytoplasmic synaptic SNARE complexes were formed by mixing syntaxin 1A (1–265, L205C), VAMP2 (1–96, Q36C), and SNAP-25B, purified and cross linked to DNA handles. An aliquot of the mixture was bound to the anti-digoxigenin antibody-coated polystyrene beads. The bead was trapped by one optical trap and brought close to a streptavidin-coated bead held by a second optical trap. The SNARE complex contained a biotinylated Avi tag added to the C-terminus of syntaxin, which could bind to the streptavidin-coated bead to form a single SNARE complex-DNA tether between the two beads. The complex was pulled to a preset constant mean force and then held at that force while keeping the trap separation fixed. The extension accompanying reversible SNARE zippering and unzipping was recorded typically at 10 kHz. Bovine α-SNAP was added into the microfluidic chamber, where a single-SNARE complex was being pulled to examine its effects on SNARE zippering.

RESULTS

HMM of single protein folding

We constructed a hidden Markov model to describe single-molecule trajectories of protein folding detected by OTs (10,24). In these experiments, a single protein or protein complex is tethered between two micron-sized polystyrene beads using one or two DNA handles (23,33). The beads are typically trapped by two optical traps formed by focused laser beams (34). By fixing the separation between the two traps, an approximately constant force can be applied to the protein, inducing protein unfolding. The unfolding is balanced by spontaneous refolding of the protein, reaching thermodynamic equilibrium in a range of forces. The reversible protein folding and unfolding causes changes in the extension of the protein and is detected by OTs through displacements of the beads in the optical traps. However, the measured extension is often obscured by measurement noise. The noise primarily arises from the Brownian motion of the beads in optical traps (19,34). We found that below 5 kHz, the noise is white and can be approximated by a Gaussian model under our experimental conditions (10,24), as is further justified in the forthcoming section. However, skewed noise distributions are often observed under different experimental conditions (11,37). In the following, we developed HMM first with a normal noise distribution (Eq. 3) and then with a skew normal noise distribution (Eq. 5). We determined the number of states in HMM based on the number of peaks in the PDF of extension (Eq. 4), as well as the transition kinetics (see the forthcoming section).

Markov models with detailed balance

A key to HMM-DB is to define a Markov model using the free energy that directly characterizes transitions of macromolecules: the free energies of stable states and the free energy of the transition states. When applied to protein folding, these energies defined a simplified energy landscape for protein folding (Fig. 1). For a system with N states, the energies form an energy matrix $\mathbf{g} = [g_{ij}]_{N \times N}$, whose

diagonal element $g_{ii} = g_i$ is the energy of the ith stable state and whose off-diagonal element $g_{ij}, \quad i \neq j$ is the absolute energy of the barrier for the transition from state i to state j. Thus, the activation energy for the transition from state i to j is $g_{ij} - g_{ii}$, as is indicated by Eq. 18. For simplicity, we chose the Nth state to be the completely unfolded protein state and set $g_N = 0$ as the energy reference for all energies. An example of the energy matrix and its associated activation energy matrix are shown in Tables 1 and S1, respectively.

These energy parameters have several advantages over the rate constants previously used (4). First, detailed balance is ensured simply by setting the energy matrix to be symmetric, or $g_{ij} = g_{ji}$ (35). This requirement is obvious from the viewpoint of the energy landscape, because both forward and backward reactions between states i and j should pass the same energy barrier or the same saddle point in the energy landscape (Fig. 1). The requirement can be rigorously proved when one expresses the state populations according to the Boltzmann distribution and the transition rates in terms of the Kramers equation, as shown in Eqs. 10 and 18, respectively. Second, our Markov model is defined by $\frac{1}{2}N(N + 1) - 1$ independent energy parameters, which are also the fitting parameters in our HMM. As a result, the number of model parameters or fitting parameters is significantly reduced compared to that of the previous model defined by an excessive number of $N(N - 1)$ rate constants. Finally, the energy landscape of protein folding can be readily derived from these energy parameters, particularly the energy landscape at zero force (32).

It must be noted that the discrete state energy differs from the energy minima or maxima in the continuous energy landscape (Fig. 1 B), which are often used interchangeably in the literature. However, the former also depends on the curvature of the potential well or barrier of the associated state in the continuous energy landscape (Eq. 14). The dependence ensures detailed balance and helps compare the free energy derived from different methods.

HMM-DB with gradient search

Given the parameters in our HMM-DB, the logarithm of likelihood (log-likelihood) $\ln L$ of observing an

TABLE 1 Matrix of Energies for the Seven-State HMM Used to Simulate the State and Extension Trajectories

State Number	1	2	3	4	5	6	7
1	<u>0</u>	7	9	8	9	8	9
2	7	<u>0</u>	9	10	7	9	9
3	9	9	<u>2</u>	11	11	11	11
4	8	10	11	<u>0</u>	8	9	10
5	9	7	11	8	<u>1</u>	9	10
6	8	9	11	9	9	<u>−1</u>	8
7	9	9	11	10	10	8	<u>0</u>

The energy of stable states is underlined. See also Fig. 2 A.

extension trajectory can be efficiently calculated using the <mark>forward-backward algorithm (1). Optimal model parameters are determined by maximizing the likelihood.</mark> We adopted a gradient search method for optimization using the MATLAB function fminunc that minimizes a target function without any constraints. We chose our target function to be the <mark>Bayesian information criterion (*BIC*),</mark> a standard criterion for model selection (36). The *BIC* is defined as $BIC = 0.5 \times N_{par} \ln(N_d) - \ln(L)$, where N_{par} is the total number of fitting parameters in the model, and N_d is the number of data points in the trajectory. Thus, we derived the gradients of the *BIC* or $\ln L$ relative to all model parameters (Supporting Materials and Methods).

HMM-DB faithfully models the simulated data

To test HMM-DB and compare with HMM-EM, we designed a seven-state hidden-Markov model mimicking the single-molecule trajectory of protein folding detected by OTs. Numbered from 1 to 7, the states had average extensions μ_i ranging from 0 to 60 nm with uniform 10 nm spacing, or $\mu_i = (i - 1) \times 10$ nm, $i = 1, \cdots, 7$, and a state-independent extension fluctuation σ_i of 4 nm. We chose an energy matrix (Table 1) such that the state populations (0.019–0.376) and transition rates (6–2479 s^{-1}; Table 2) fell into large ranges detectable by OTs. To simulate the corresponding extension trajectory, we first generated a time-dependent state trajectory based on the Markov model (Table S2) and then added noise based on the state-dependent Gaussian distribution (Eq. 3).

Fig. 2 *A* shows snapshots of the simulated state trajectory (*red circles*) and extension trajectory (*gray line*) with a total time length $T_d = 10$ s and a sampling time $\Delta t = 0.2$ ms. The PDF of the whole extension trajectory exhibited seven overlapping peaks with varying heights (Fig. 2 *B*, *circles*). We then maximized the likelihood of observing the simulated extension trajectory using seven-state HMM-DB. The model had a total of 41 fitting parameters, including six stable state energies ($g_7 = 0$), 21 transition state energies, seven average state extensions, and seven state fluctuations. Fitting to 50,000 data points, the gradient search quickly maximized the log-likelihood (Fig. 2 *C*) with all fitting parameters converging to unique values, independent of their initial values. Correspondingly, the gradients of the log-likelihood relative to all fitting parameters were approximately zero at the converged values (Fig. 2 *D*). Varying each parameter around its converged value increased the negative log-likelihood quadratically, which revealed the fitting error for each parameter (ξ in Fig. 2 *D*). We concluded that the converged parameters were the best-fit parameters that globally minimized the BIC.

Our HMM-DB faithfully recovered the input model parameters. First, the idealized state trajectory derived from HMM-DB (Fig. 2 *A*, *cyan crosses*) matched the simulated input state trajectory (*red circles*) well, with only 4.25% of time points showing state mismatches (data points indicated by *blue arrows*). Second, the extension PDF calculated from the best-fit parameters (Fig. 2 *B*, *solid red curve*) overlapped the PDF of the simulated extension. Third, the state populations and lifetimes were close to the model inputs (Tables S3 and S4). Fourth, and finally, the best-fit parameters generally matched the input model parameters well, as is shown by their small differences (Fig. 3; Table S5). Close inspection revealed that the energies, extensions, and fluctuations of all stable states exhibited smaller deviations from the corresponding input values, even for the least populated state 3. Accordingly, these parameters also showed smaller fitting errors (Table S5). In contrast, the energies of transition states showed relatively larger deviations, especially for the transitions starting from state 3. These observations are justified by the fact that the fitting accuracy was primarily determined by the numbers of associated transition events (Table S2). In conclusion, our HMM-DB faithfully modeled the simulated trajectory.

Comparison between HMM-DB and HMM-EM and effects of detailed balance

To understand the effects of detailed balance on HMM, we analyzed the same extension trajectory described above using HMM-EM. Without the constraints of detailed balance, the HMM-EM contained 56 fitting parameters. To compare the results of HMM-EM with those of HMM-DB, we computed the apparent energies from the best-fit transition probabilities (Eq. 19). Because the results of HMM-EM may not satisfy detailed balance, the derived energy g_{ij} may not equal g_{ji}. Thus, we computed their averages $\overline{g}_{ij}$ and differences Δ_{ij}. We compared the average $\overline{g}_{ij}$ to the corresponding energy $g_{ij} = g_{ji}$ from HMM-DB (Table S5) and used Δ_{ij} to quantify the degree that HMM-EM violates detailed balance (Eqs. 20–23). HMM-DB and HMM-EM recovered equally well the input state extensions (Fig. 2 *A*; Table S5), fluctuations, energies, populations (Table S3), trajectory (Fig. 2 *A*), and the extension PDF (Fig. 2 *B*). In addition, HMM-DB shared the high computational efficiency of HMM-EM, both of which took <25 s to reach the preset convergence criterion of 0.001 for the

TABLE 2 Matrix of the Transition Rate Constants Calculated from the Energy Matrix Shown in Table 1 Using Eq. 18

State Number	1	2	3	4	5	6	7
1	−1953	912	123	335	123	335	123
2	912	−2239	123	45	912	123	123
3	912	912	−2317	123	123	123	45
4	335	45	17	−902	335	123	45
5	335	2479	45	912	−4230	335	123
6	123	45	6	45	45	−389	123
7	123	123	17	45	45	335	−690

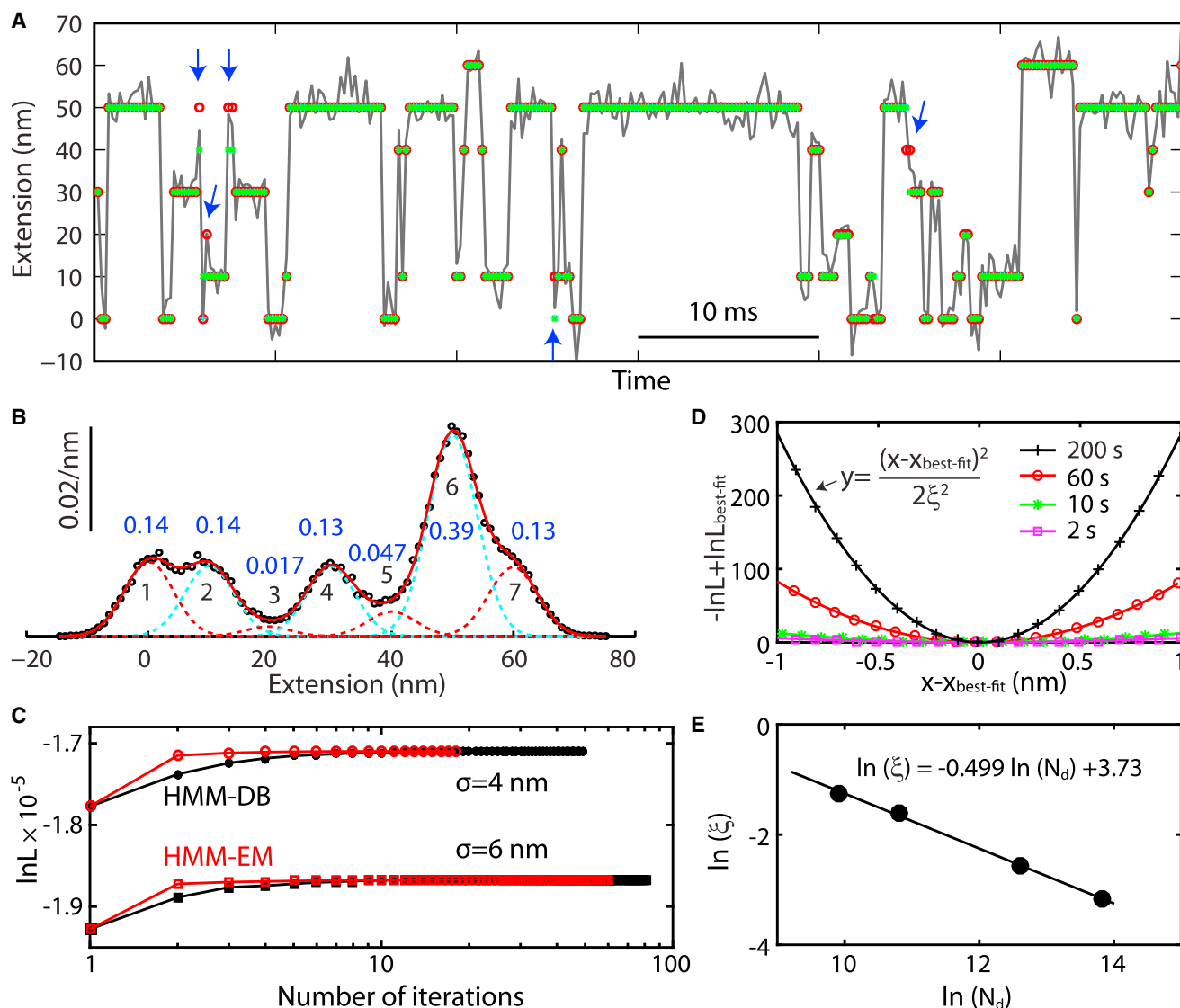Transition rate constants are in unit s^{-1}.

FIGURE 2 HMM-DB better recovers the parameters of a seven-state hidden-Markov model from simulated extension data than the standard HMM-EM. (*A*) Simulated extension-time trajectory with noise (*gray line*) and without noise (*red circles*) and the idealized state trajectory derived from HMM-DB (*cyan*, +) and from HMM-EM (*green*, ×). The two idealized state transitions overlap well, with only 0.3% mismatched data points, and deviated from the input state transitions with 4.3% mismatches (indicated by *blue arrows*). (*B*) PDF of the simulated extension partially shown in (*A*) (*black circles*) and its best-fit values by HMM-DB (*red curve*). Average extensions and fluctuations of all seven states were faithfully recovered, as are indicated by individual Gaussian functions shown as dashed curves. The associated state numbers (1–7 in *black*) and populations (*blue numbers*) are labeled. The PDF derived from HMM-EM overlaps the one derived from HMM-DB, and thus is not shown. (*C*) Log-likelihood as a function of the iteration step during maximization by HMM-DB (*black*) and HMM-EM (*red*) for the simulated 10 s extension trajectories with two noise levels. (*D*) Relative log-likelihood ($-\ln L + \ln L_{\text{best-fit}}$, *symbols*) as a function of the selected fitting parameter ($x = \mu_3$) around its best-fit values ($x_{\text{best-fit}}$) for extension trajectories with different time lengths (2–200 s). The relative log-likelihood was fit by a quadratic function (*inset*) to obtain the fitting error ξ, with the best fits shown as curves. (*E*) Fitting error ξ as a function of the number of data points N_d (*solid circle*) derived from the data shown in (*D*) and its best fit by a straight line (*line*). To see this figure in color, go online.

difference of log-likelihood between consecutive iterations of optimization (Fig. 2 *C*). Although HMM-DB generally needed more iterations to converge than HMM-EM, HMM-DB spent less CPU time per iteration. However, HMM-DB outperformed HMM-EM when noisy trajectories were analyzed. When the state fluctuation was increased from 4 to 6 nm, without other changes in the model, HMM-DB required 36 s to fit the simulated extension trajectory, while HMM-EM needed 54 s (Fig. 2 *C*). In this case, the gradient

search converged faster than the EM algorithm near the shallow minimum of the negative log-likelihood (4). The efficient gradient search required the derived analytical gradients with respect to all fitting parameters (Supporting Materials and Methods). Replacing the analytical gradients with their finite-difference approximations built in the fminunc MATLAB function increased the CPU time to ~2200 s.

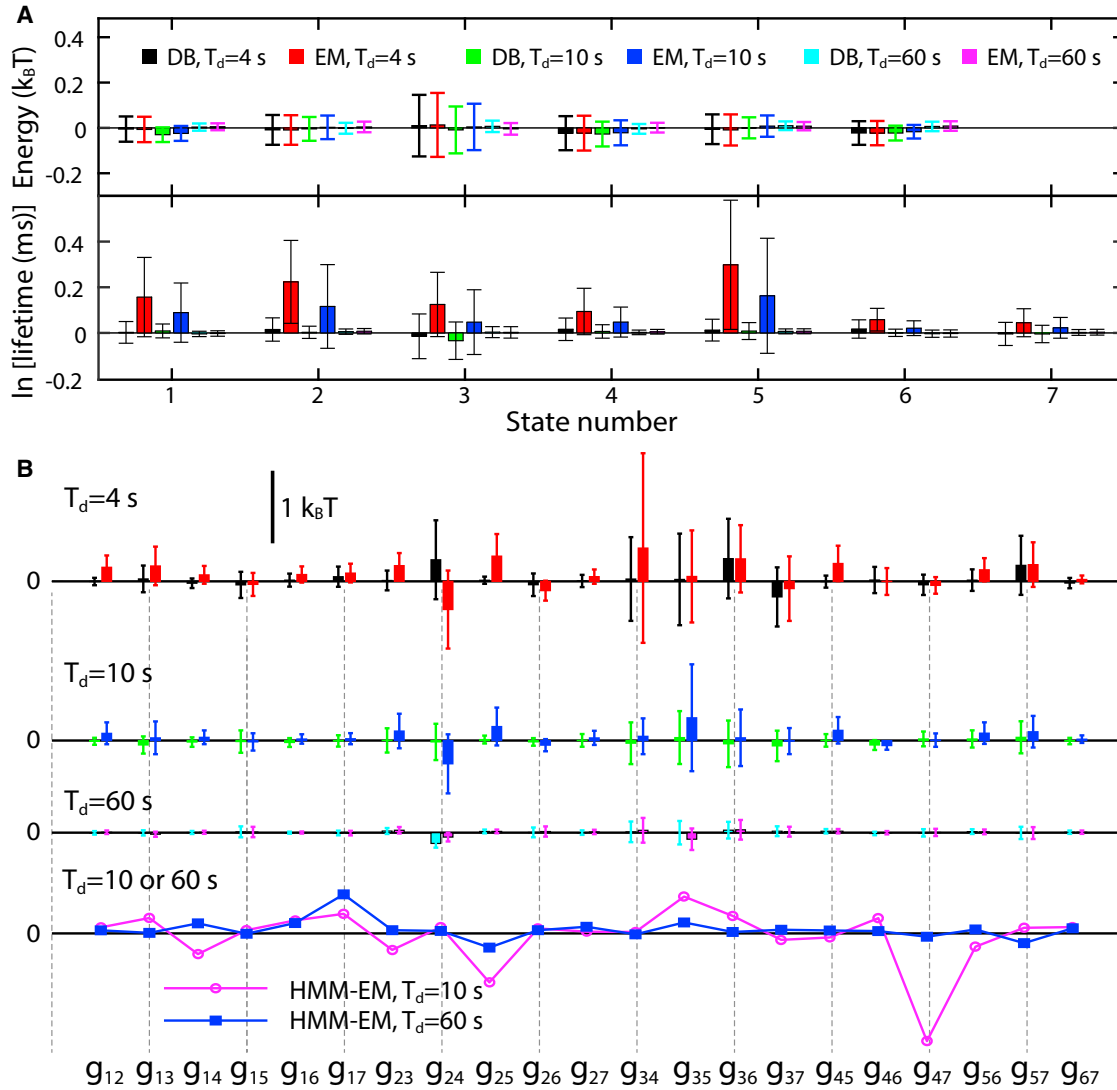HMM-DB more accurately fit the transition rates than HMM-EM, especially for short trajectories. Due to the

FIGURE 3 HMM-DB better models short extension trajectories than does HMM-EM. (A) Deviation of the best-fit state energy (*top panel*) and lifetimes (*bottom panel*) from their corresponding input model values for the simulated extension trajectories with time lengths (T_d) of 4, 10, and 60 s. (B) Deviation of the best-fit transition state energy (*top three panels*) derived from HMM-DB and HMM-EM and the energy difference (Δ_{ij}) derived from HMM-EM. Different bars or symbols are colored the same as in (A).

stochastic nature and finite data length of the simulation, the best-fit parameters often showed small systematic deviations from the input parameters. To minimize such systematic error and better compare HMM-DB and HMM-EM, we repeated the above simulation of the same HMM model and subsequent HMM analysis 10 times for each of simulation time lengths of 4, 10, and 60 s. We compared the means (or mean deviations) and standard deviations (or variations) of the best-fit parameter values relative to their corresponding input values. Consistent with above results, HMM-DB and HMM-EM demonstrated comparable accuracy in determining the state energies (Fig. 3 A, *top panel*), positions, and fluctuations (Fig. S1), as are shown by their similar mean deviations and variations. However, for 4 and 10 s trajectories, HMM-DB consistently yielded smaller mean devia-

tions and variations for the lifetimes of all seven states (Fig. 3 A, *bottom panel*) and the energies of most transition states (Fig. 3 B, the *first two panels*) than HMM-EM did. For the 4 s trajectory, HMM-DB revealed much more accurate fitting results than HMM-EM. In contrast, the fitting accuracy of both HMM methods approaches a similar level for the 60 s trajectory (Fig. 3; Tables S4 and S5). Finally, for both HMM methods, the fitting accuracy increased as the data length increased (Fig. 3), as is predicted by HMM theory (1,4). Particularly, the fitting error ξ decreased with the number of data points N_d in a manner $\xi \propto 1/\sqrt{N_d}$, which is proved by the fitting error for the average extension of state 3 (Fig. 2, *D* and *E*). These comparisons demonstrated that HMM-DB more accurately derived the transition kinetics from short trajectories than HMM-EM did.

We speculated that detailed balance in HMM-DB reduced overfitting of short extension trajectories, leading to a higher fitting accuracy than in HMM-EM. At a time length of 10 s, some transitions only appeared 17 times (Table S2). As a result, HMM-EM might overfit the extension trajectory given its excessive number of fitting parameters. To investigate the effect of detailed balance on the improved accuracy of HMM-DB, we examined how HMM-EM might violate detailed balance. We plotted the differences between the energy barriers Δ_{ij} for all pairs of forward and backward transitions (Fig. 3 B, *bottom panel*). If detailed balance holds true, all energy differences should be zero or $\Delta_{ij} = 0$, for all $i \neq j$, $i, j = 1, \cdots, N$. However, we found that many energy differences Δ_{ij} significantly deviated from zero for the 10 s trajectory, revealing that the results of HMM-EM indeed violated detailed balance. Consequently, steady-state probability flux existed in various loops in the reaction network. In contrast, an increase in the time length to 60 s minimized the energy differences of the transition states, showing approximate detailed balance. Compared to the relative fitting accuracy of the two HMM methods on the two trajectories, we concluded that detailed balance conferred an advantage for HMM-DB over HMM-EM to accurately determine the kinetics of a hidden Markov model.

HMM-DB with a skew noise distribution

An accurate statistical model for the measurement noise is crucial for HMM of single-molecule trajectories and derivation of folding energy landscapes of macromolecules (10,11,13,28,32). Depending on experimental conditions, the extension fluctuation has been described by either a normal distribution or a skew distribution (10,11,13,37). Correspondingly, we replaced the normal distribution described above with a skew distribution. A skew noise distribution arises due to the nonlinear mechanical response of the DNA handle or optical traps (10,13,37). Several mathematical functions modified from a Gaussian function have been used to fit the skew extension histograms (11,37). However, these functions cannot be used in HMM as a PDF, because their normalized analytical expressions are not available. We adopted the skew normal distribution to describe the skew noise distribution measured by optical tweezers (30) (Eq. 5), which contains three parameters: the location μ, the scaling factor σ, and the shape ζ. The new shape parameter characterizes the skewness of the distribution: the tail of the distribution is located on the right side when $\zeta > 0$ and on the left side when $\zeta < 0$. When $\zeta = 0$, the distribution is reduced to the normal distribution. We chose the skew normal distribution in our HMM, because the corresponding PDF has been well studied in statistics and has closed-form mean, standard deviation, and skewness (Eqs. 7–9), which can be utilized to directly derive the best-fit PDF for the measured extension fluctuation around a single protein state (Supporting Materials

and Methods). Furthermore, we could derive the gradients of the log-likelihood with respect to the three parameters in the skew-normal distribution.

To test the HMM-DB with a skew normal noise distribution, we simulated a 30 s extension trajectory based on a three-state hidden-Markov model (Fig. 4 A). Despite the distinct shapes or skewness of the noise distributions for the three states, our HMM-DB accurately recovered the input parameters, with an average absolute difference ~0.073 for all fitting parameters. As a result, the idealized state transition matched the input state transition, with only 0.01% mismatches (Fig. 4 A), and the PDFs derived from HMM-DB overlapped the corresponding PDFs of the simulated extension (Fig. 4 B). Note that the mean $\bar{x}$ and the standard deviation ω of a skew normal distribution depends on the shape parameter ζ (Eq. 7). To better compare the idealized state transitions with different noise distributions, we plotted the mean of state extensions to show the idealized state transition (Fig. 4 A). The comparison validated our HMM of extension trajectories with skew normal noise distributions.
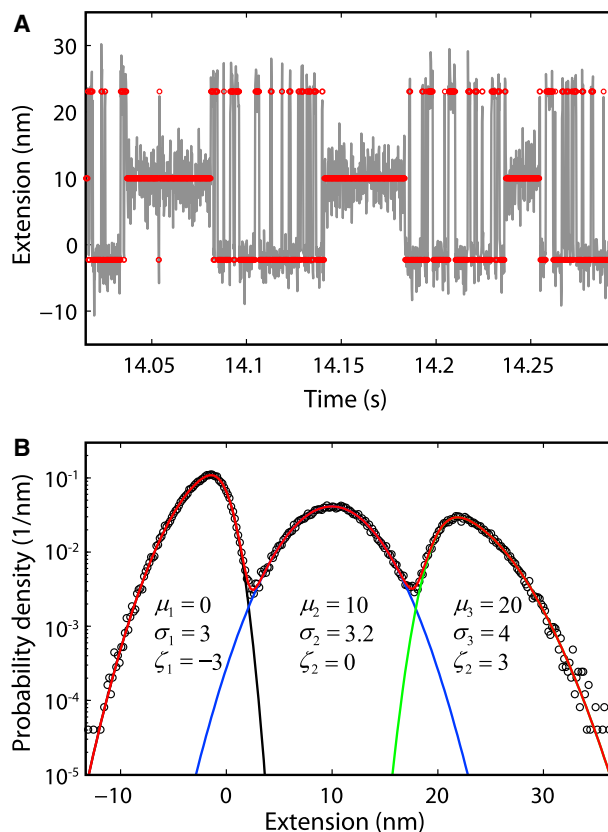

FIGURE 4 HMM-DB accurately models the simulated extension trajectory with a skew noise distribution. (A) Snapshots of the simulated extension-time trajectory (*gray*) and the idealized state transitions (*red circle*) based on the three-state hidden Markov model with energies $g_1 = 1$, $g_2 = -0.5$, $g_{12} = 11$, $g_{13} = 7$, and $g_{23} = 10$. (B) Input parameter values for the skew normal distributions. PDF of the extension in (A) (*circle*) and its best fit by HMM-DB (*red curve*). The best-fit PDFs for individual states are shown in black, blue, and green lines for states 1, 2, and 3, respectively. To see this figure in color, go online.

Small skewness of the measured noise distribution

To determine the noise distribution under our experimental condition, we held the 2260 bp DNA handle at different constant mean forces and measured its extension for over 10 s. Fig. 5 shows a representative distribution of the extension obtained at 12.7 pN force. We fit the extension distribution by both skew normal and normal noise distributions using a maximum-likelihood method. Both distributions fit the extension distribution well. Accordingly, the fitting revealed a shape parameter ζ of -0.58, indicating a small skewness of the noise distribution ($\gamma = -0.04$, Eq. 9). The conclusion was supported by our extension measurements and analyses, which showed an average shape $\zeta = -0.6$ (± 0.1, standard deviation, $N = 12$). In addition, the shape did not significantly change with force and the bandwidth of the extension data analyzed. The location and the scaling factors were generally greater than the corresponding parameters μ and σ, respectively, in the normal distribution (Fig. 5, inset). However, the average extensions and their standard deviations of the skew normal and normal distributions were almost identical. In conclusion, the noise distribution was slightly negatively skewed under our experimental conditions. We noticed that significantly skewer noise distributions were reported under different experimental conditions (11,37). In our experiments, we used ~2 μm diameter beads and a trap stiffness of ~0.2 pN/nm for both optical traps, while in previous experiments, shorter DNA handles (~1 kbp), smaller beads (~1 μm diameter), or stiffer traps (>0.3 pN/nm) were utilized (11,37).

The maximum log-likelihood calculated using a skew normal distribution is 23, higher than that obtained using a normal distribution for the 40 s extension trajectory (Fig. 5, inset). The difference is significant, even after sub-tracting the penalty of one additional parameter for the skew normal distribution (~6) based on the BIC. The possible impact of this difference on HMM was examined in the following section.

HMM-DB and HMM-EM model SNARE zippering equally well

Soluble n-ethylmaleimide-sensitive factor attachment protein (SNAP) receptors (SNAREs) couple their stepwise folding and assembly to membrane fusion like a zipper (38). Using OTs, we measured the folding energy and kinetics of the synaptic SNARE complex responsible for fusion of synaptic vesicles to presynaptic plasma membranes (24,33,39). The SNARE zipper comprises two pieces, the t-SNARE complex and the v-SNARE, which are located on the target plasma membrane and the vesicle membrane, respectively. The binary t-SNARE complex folds into a three-helix bundle that serves as a rigid template for stepwise v-SNARE zippering (33). The fully zippered SNARE complex forms a parallel four-helix bundle (40). We cross-linked the cytoplasmic t- and v-SNAREs at their N-termini and pulled on their C-termini using OTs (26,33). At an approximately constant force of 16.5 pN, the SNARE complex rapidly folded and unfolded among three states, as is shown by the extension-time trajectory (Fig. 6, A, top gray trace, and B).

We analyzed the 26.4 s trajectory at a bandwidth of 5 kHz (or $dt = 0.2$ ms) using three methods: HMM-DB and HMM-EM with a normal noise distribution, and HMM-DB with a skew normal distribution. The three methods modeled the trajectory equally well, yielding overlapping idealized state transitions (Fig. 6 A, colored symbols), close best-fit parameters (Figs. 6 C and S2; Table S6), and PDFs (Fig. 6 D, top panel). Thus, including the skew noise distribution in our HMM-DB minimally affected the transition kinetics, and showed that the measured noise distribution could be well approximated by a normal distribution. Both HMM-DB methods showed that SNARE zippering was sequential, in which the partially zippered state 3 mediated transitions between the zippered state 2 and the unzipped state 4, with no direct transitions between states 4 and 2 (Fig. 6 C). In contrast, HMM-EM exhibited minor direct transitions between the two states, with $q_{42} = 0.23$ s^{-1} and $q_{24} = 0.07$ s^{-1}, which led to a degree of imbalance of 0.28 for the transitions between the two states. In conclusion, a normal distribution modeled the noise in our experiments well and detailed balance helped inhibit the nonsequential SNARE transitions modeled by HMM-DB.

HMM-DB better modeled SNARE zippering with degenerate states

We then added 5 μM SNAP (or specifically α-SNAP) into the solution to investigate how SNAP regulated SNARE
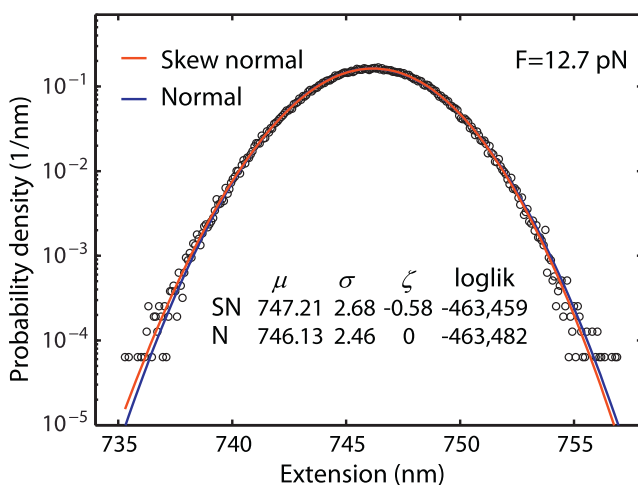

FIGURE 5 PDF of the measured extension fluctuation (circles) is slightly skewed. The PDF was fit by both the skew normal (SN) distribution (red curve) and the normal (N) distribution (blue curve) using maximum likelihood estimation; (inset) best-fit parameters and their associated maximum likelihood. To see this figure in color, go online.
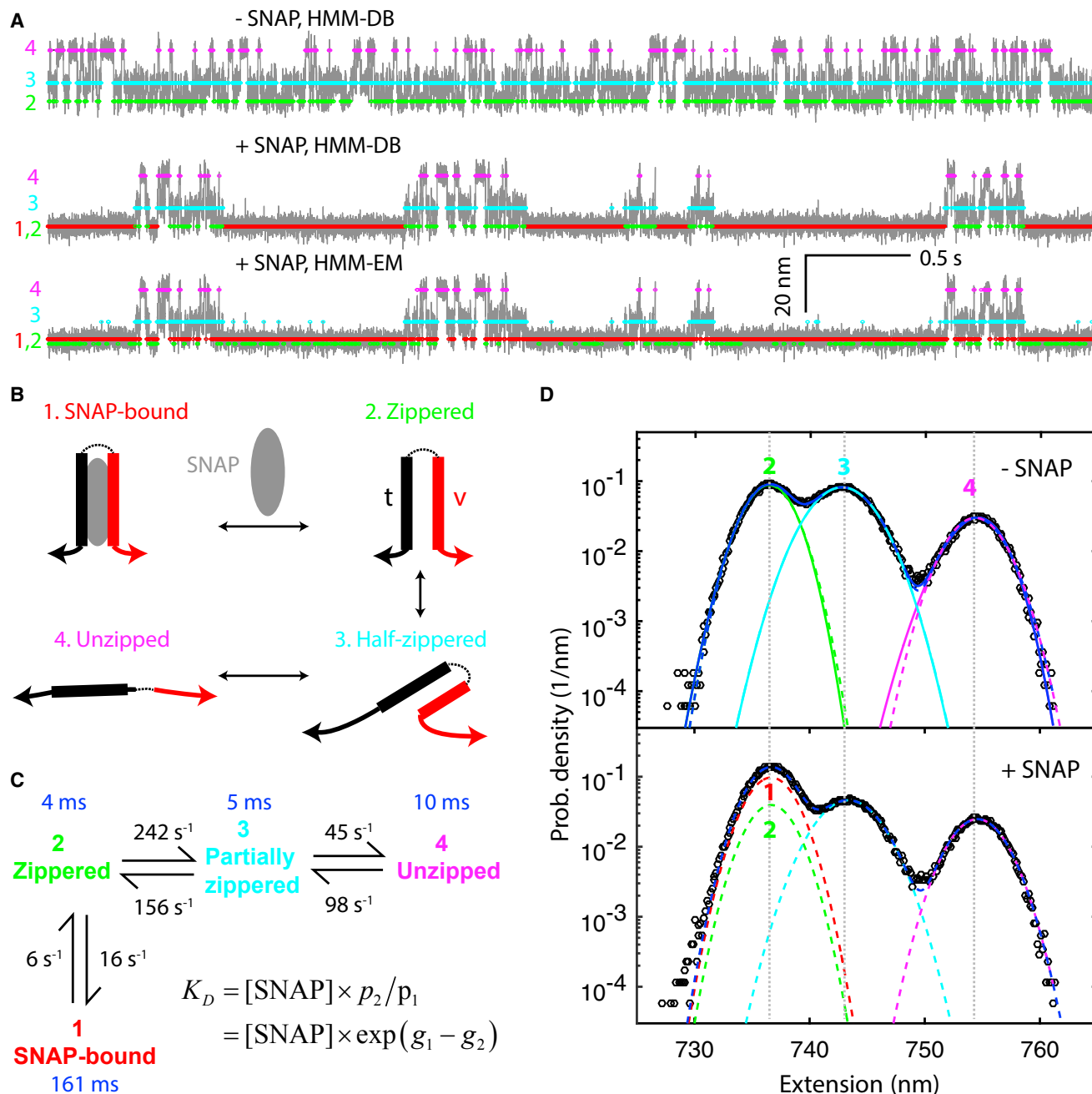
FIGURE 6 HMM-DB models the extension trajectory of SNARE zippering and SNAP binding containing degenerate states better than HMM-EM. (*A*) Extension-time trajectories (*gray lines*) of a single SNARE complex under a constant mean force of 16.5 pN in the absence (− SNAP) and presence (+ SNAP) of SNAP in the solution. For the extension trajectory in the absence of SNAP, the idealized state transitions (*colored symbols*) that were derived from HMM-DB and HMM-EM with a normal distribution and HMM-DB with a skew normal distribution were found to be indistinguishable. (*B*) Schematic model of SNARE zippering and SNAP binding. The SNARE complex was pulled from the site and direction indicated by arrows. (*Gray oval*) SNAP. (*C*) Rates (*black numbers*), state lifetimes (*blue numbers*), and pathway of SNARE zippering and SNAP binding derived from HMM-DB. (*Inset*) Dissociation constant was calculated using this formula; the transition rates and state lifetimes of SNARE zippering in the absence of SNAP are close to the corresponding values in the presence of SNAP. (*D*) PDFs of the extensions in (*A*) (*symbols*) and their best-fit values (*blue lines*) derived from HMM-DB. The extension in the absence of SNAP was fit by three-state HMM using both the normal (*dashed lines*) and the skew normal (*solid lines*) noise distributions, while the extension in the presence of SNAP was fit using a normal distribution. The best-fit PDFs for individual states are also shown. To see this figure in color, go online.

zippering under otherwise identical conditions. As was recently reported in Ma et al. (26), SNAP binding stabilized the zippered SNARE state, leading to a new long-dwelled

state 1 (Fig. 6, *A*, *middle gray trace*, and *B*, from states 2 to 1). Our goal here was to reveal the SNAP binding mechanism and kinetics using HMM. A major challenge to

analyze the trajectory was the equal extensions of the two degenerate states, the SNAP-bound state 1 and the free zipped SNARE state 2.

We first analyzed the 25.28 s trajectory using four-state HMM-DB. To minimize the number of parameters in the model, we used the normal noise distribution and constrained the extensions of the two degenerate states 1 and 2 to be equal. Initial HMM showed that direct transitions between states 1 and 4 and between states 2 and 4 were negligible. Thus, to further simplify the model, we inhibited these transitions, leading to the sequential reaction scheme shown in Fig. 6 C. HMM-DB fit the extension trajectory and its PDF well (Figs. 6, A, *middle colored symbols*, and D, and S3). Particularly, HMM-DB correctly separated the SNAP-bound state 1 and the free zipped state 2 based on their distinct lifetimes (Fig. 6 C, *blue numbers*). The average lifetime of the SNAP-bound state was 161 ms, compared to 4 ms for the free zipped state. The latter lifetime, or $\tau_2 = 1/(q_{23} + q_{21}) = 4$ ms, was only slightly shorter than the 4.3 ms lifetime of the same state in the absence of SNAP, because the unfolding rate constant $q_{23} = 242$ s^{-1} was much greater than the SNAP binding rate constant $q_{21} = 15.5$ s^{-1}. Similarly, SNAP only minimally altered the lifetimes of the two other SNARE states 3 and 4. In particular, SNAP did not directly bind the partially zipped state 3, as was revealed by the high energy barrier ($>23\,k_BT$) derived for the transitions between states 1 and 3 (Fig. S3). This finding suggests that SNAP did not induce folding of the SNARE complex. Instead, SNAP only targeted the zipped state 2 to bind the SNARE complex, indicating a SNAP-binding mechanism by conformational search (26). Accordingly, we found that an increase in the SNAP concentration only increased the SNAP binding rate constant q_{21}, but did not change other rate constants (26). This observation confirmed our model derived from HMM-DB (Fig. 6 C).

HMM-EM of the same extension trajectory converged to different sets of parameter values or HMM models, depending on the initial parameters. All these models significantly differed from the unique model derived from HMM-DB described above. The model from HMM-EM with the lowest BIC is shown in Fig. 6 A (*bottom panel*). Although HMM-EM correctly identified states 3 and 4, it failed to separate the two degenerate states: the derived states had different extensions but close lifetimes (~2 ms). In addition, rapid transitions were found between the SNAP-bound state and the partially zipped state ($q_{13} = 97$ s^{-1}, $q_{31} = 132$ s^{-1}). Finally, the model dramatically changed as HMM-EM was performed using the extension trajectory filtered to 1 kHz (or $dt = 1$ ms), while the results from HMM-DB did not significantly change. We concluded that this and other models (Fig. S4) derived from HMM-EM were artifacts of local optimum and overfitting due to excessive model parameters in HMM-EM. Interestingly, when using the best-fit model parameters from HMM-DB as initial

fitting parameters, we found that HMM-EM converged to the model derived by HMM-DB. In this case, the sequential SNARE folding and SNAP binding kinetics contained in the input model was maintained in parameter optimization by HMM-EM (1). As a result, detailed balance was automatically satisfied. This finding supports the role of detailed balance in model identifiability in degenerate hidden Markov models.

Deriving the energy landscape of protein folding

A reasonable hidden Markov model should also account for its dependence on implicit model parameters, such as the ligand concentration and force. The dependence in turn puts a strong constraint on the HMM model. Here we further justified our model of SNARE zippering and SNAP binding derived from HMM-DB by its dependence on force. We measured the extension trajectories of SNARE zippering in the presence of 5 μM SNAP at different forces (26). We similarly analyzed these trajectories using HMM-DB, leading to the force-dependent energy shown in Fig. 7 A. The energy was the total relative energy of the dumbbell system when
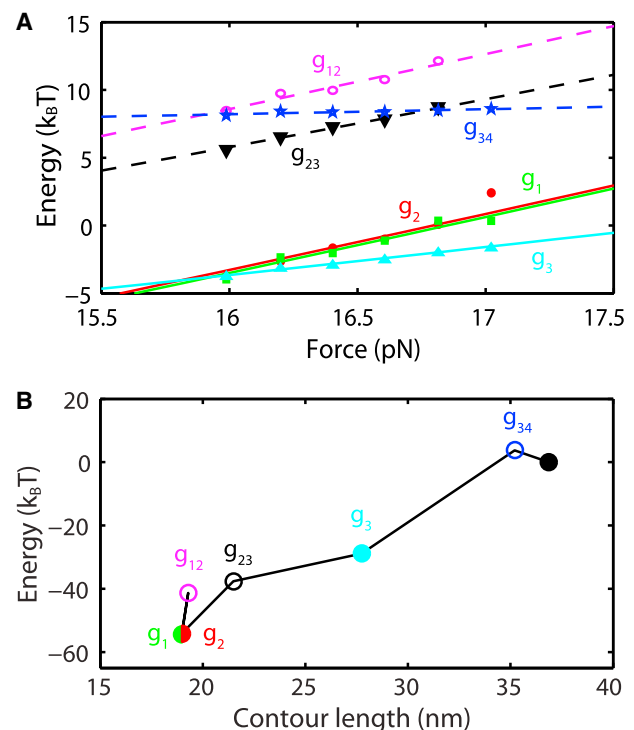
FIGURE 7 Deriving the simplified protein folding energy landscape from the force-dependent state energies calculated by HMM-DB. (A) Force-dependent energies (*symbols*) and their best-fit values (*lines*) by a theoretical model for SNARE zippering and SNAP binding (32). The energies of different states at each force are defined relative to the energy of the unzipped state at that force. (B) Simplified energy landscape of SNARE zippering and SNAP binding at zero force derived from the model fitting in (A). The contour length of the unfolded and stretched polypeptide also includes the contribution from the spacer sequence added to the SNARE complex required to attach the complex to the beads (26).

the protein stayed in a specific state, including the energy of the folded protein portion, the entropic energy of the unfolded polypeptide portion and of the DNA handle, and the potential energy of the beads in optical traps. Whereas the energy of the folded protein portion in a protein state barely changes with force, the entropic energy increases as force increases. We calculated the total energy as a function of force and fit the calculations to the corresponding energy derived from HMM (32). The fitting parameters consisted of the energies of stable and transition states of the protein at zero force (g_{ij}) and their corresponding contour lengths of the unfolded polypeptide portions (l_{ij}). Here the contour length was chosen as a reaction coordinate to characterize the protein folding pathway. Thus, these best-fit parameters constitute a simplified energy landscape of protein folding at zero force (Fig. 1 B).

The energies of all stable and transition states relative to the energy of the unzipped state increased linearly as force increased in the force range tested. This observation is expected (10,32), because force approximately increased the relative energy in proportion to their contour length differences, i.e., $g_{ij}(F) \propto F(l_{NN} - l_{ij})$, where the contour length of the unfolded state l_{NN} is generally known (10). Thus, the slope of the energy versus force curve is roughly proportional to the contour length of the unfolded polypeptide in that protein state. However, to extrapolate the energy to zero force, we utilized a nonlinear model of the force-dependent energy and fit the model to the experimental measurements by nonlinear least-squares fitting (32). Our model fit the measurements well (Fig. 7 A) and thus yielded the energy landscape for SNARE zippering and SNAP binding

(Fig. 7 B). Particularly, the folding energy of the SNAP-bound state and the free zippered SNARE state were $-54.3 \ k_BT$ and $-54.1 \ k_BT$, respectively. The energy difference $(g_1 - g_2)$ gave the SNAP-SNARE association energy of $-0.2 \ k_BT$ at 5 μM SNAP concentration. The energy is equivalent to a dissociation constant K_d of 4 μM or an association energy of $\sim-12 \ k_BT$ under standard conditions (Fig. 6 C, inset), which is close to a recent measurement of 1.5 μM (41). Results from direct fitting to the energy are consistent with those from fitting to state populations and transition rates, as we recently reported in Ma et al. (26) and Rebane et al. (32). However, fitting to energy is simpler because the energy parameters are independent.

HMM-DB with baseline correction

The reduced number of model parameters in our HMM-DB compared with HMM-EM allowed us to incorporate baseline correction without overfitting the extension trajectories. Fig. 8 A shows an extension trajectory of SNARE zippering with slow baseline drift. The baseline was represented by a series of piecewise-linear functions (9), whose coefficients were optimized by minimizing the BIC using the gradient search method (Supporting Materials and Methods). The best-fit baseline using 10 piecewise functions (Fig. 8 B) matched the contour of the measured extension trajectory and was subtracted from the latter to generate the corrected extension trajectory (Fig. 8, C and D). We compared HMM-DB of the extensions without and with baseline correction. The fitting for the two trajectories were generally close, including the best-fit parameters and the idealized
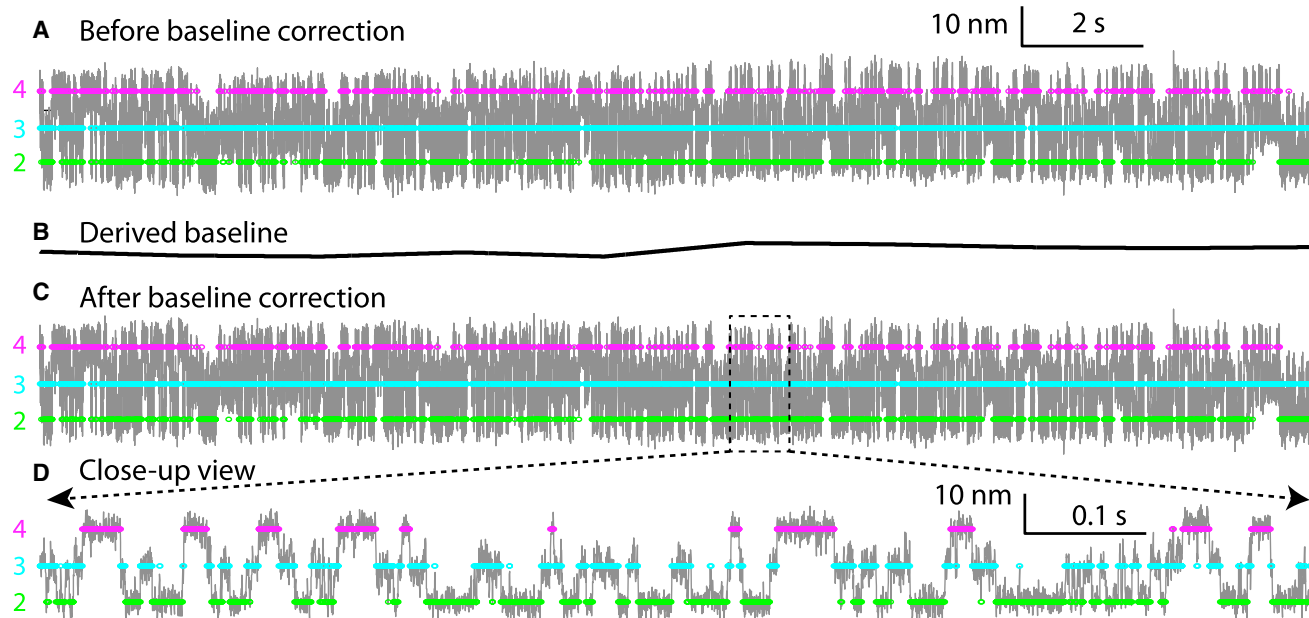
FIGURE 8 Correction of the baseline drift by HMM-DB. Extension-time trajectories that show three-state SNARE zippering under a constant mean force of 17.8 pN before (A) and after (C and D) baseline correction and the derived baseline (B). A close-up view of the region in the dashed box is shown in (D). Idealized state transitions are shown in colored symbols, with their state numbers (Fig. 6 B) shown on the left. To see this figure in color, go online.

state transitions (Fig. 8, *A*, *C*, and *D*). However, the extension PDFs revealed differences (Fig. 9). The baseline-corrected extension showed increased separation between the average extensions of the three states, with 0.62 nm upshift for the unzipped state. In addition, baseline correction reduced the extension fluctuations around all three states by ~10%, leading to sharper peaks in the extension PDF. Because the extension of a state is closely related to its derived structure (32), this example demonstrated that HMM with baseline correction improves the accuracy of structures derived from protein folding studies using OTs.

DISCUSSION

Compared with previous HMM based on gradient search (4), an important improvement introduced by us was the use of energies as model parameters. These parameters ensure detailed balance and serve as independent fitting parameters in HMM. Our method is especially useful in protein folding studies, in which deriving the energy landscape of protein folding is of key importance. We obtained a simplified energy landscape that complied with the Boltzmann distribution, the Kramers rate theory, and detailed balance, and related the discrete energy landscape to the continuous energy landscape. We demonstrated that HMM-DB more accurately modeled single-molecule trajectories with limited data points or degenerate states than HMM-EM. When applied to experimental data of SNARE zippering and SNAP binding that contained degenerate states, HMM-DB revealed a reasonable model that was supported by its dependence on force and SNAP concentration, whereas HMM-EM did not. The comparisons demonstrated the advantage of HMM-DB in analysis of single-molecule trajectories involving transitions at thermodynamic equilibrium.

Degenerate hidden Markov models often lack sufficient information to identify all parameters in the models, leading to model nonidentifiability (25,42,43). In other words, two or more distinct sets of parameters may fit the data equally well. In conjunction with overfitting, we believe that such model nonidentifiability caused convergence of HMM-EM to different SNARE zippering and SNAP binding models observed by us. Detailed balance and the dependence of force or protein concentrations add more constraints to the model, making it possible to determine all model parameters (43).

To accommodate different experimental conditions, we developed HMM-DB with both normal and skew normal noise distributions. Although a discrete noise distribution can be derived from HMM without assuming any functional form of the distribution (14), a continuous functional description of the noise distribution enables faster and more robust HMM. Whereas many methods expand the applicability of HMM by increasing the number of fitting parameters (7,12,14), our HMM-DB reduces the fitting parameters to a minimum number and thus may serve as one of the first methods to analyze complex reaction networks under thermal equilibrium. Compared to other methods for the analysis of single-molecule trajectories (17,28), HMM requires that the noise in the model be completely uncorrelated, which may limit the temporal resolution of the data analyzed by HMM. The high-resolution data from optical tweezers should be properly filtered to remove any extension correlation caused by bead relaxation in optical traps, typically to a bandwidth below the corner frequency of the beads under tension (24). Although our applications of HMM-DB were focused on protein folding, our method should have wide applications to systems at thermodynamic equilibrium.

SUPPORTING MATERIAL

Supporting Materials and Methods, four figures, and six tables are available at http://www.biophysj.org/biophysj/supplemental/S0006-3495(16)30879-7.

AUTHOR CONTRIBUTIONS

Y.Z., J.J., and A.A.R. designed the research; Y.Z., J.J., and A.A.R. performed the calculations; and Y.Z., J.J., and A.A.R. wrote the article.

ACKNOWLEDGMENTS

We thank Frederick Sigworth and Attila Szabo for discussion.

FIGURE 9 Comparison of the PDFs of the extensions (*symbols*) before (*black*) and after (*red*) baseline correction and their best-fit values by HMM-DB (*lines*). To see this figure in color, go online.

REFERENCES

1. Rabiner, L. R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE.* 77:257–286.
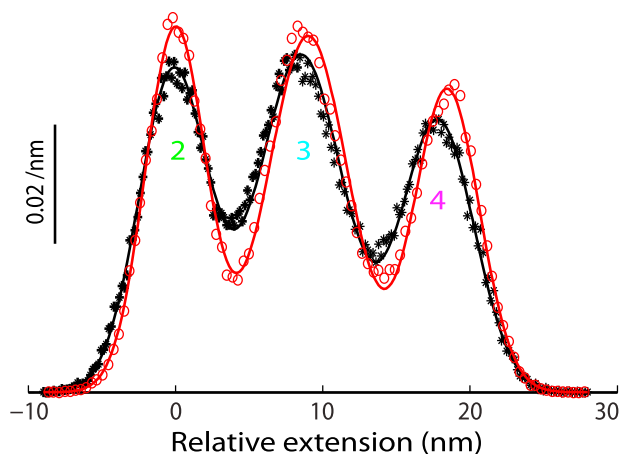
2. Eddy, S. R. 2004. What is a hidden Markov model? *Nat. Biotechnol.* 22:1315–1316.

3. Sigworth, F. J. 2016. Principles of cryo-EM single-particle image processing. *Microscopy (Oxf.).* 65:57–67.

4. Qin, F., A. Auerbach, and F. Sachs. 2000. A direct optimization approach to hidden Markov modeling for single channel kinetics. *Biophys. J.* 79:1915–1927.

5. Qin, F. 2014. Principles of single-channel kinetic analysis. *Methods Mol. Biol.* 1183:371–399.

6. McKinney, S. A., C. Joo, and T. Ha. 2006. Analysis of single-molecule FRET trajectories using hidden Markov modeling. *Biophys. J.* 91:1941–1951.

7. van de Meent, J. W., J. E. Bronson, …, R. L. Gonzalez, Jr. 2014. Empirical Bayes methods enable advanced population-level analyses of single-molecule FRET experiments. *Biophys. J.* 106:1327–1337.

8. Müllner, F. E., S. Syed, …, F. J. Sigworth. 2010. Improved hidden Markov models for molecular motors, part 1: basic theory. *Biophys. J.* 99:3684–3695.

9. Syed, S., F. E. Müllner, …, F. J. Sigworth. 2010. Improved hidden Markov models for molecular motors, part 2: extensions and application to experimental data. *Biophys. J.* 99:3696–3703.

10. Gao, Y., G. Sirinakis, and Y. Zhang. 2011. Highly anisotropic stability and folding kinetics of a single coiled coil protein under mechanical tension. *J. Am. Chem. Soc.* 133:12749–12757.

11. Stigler, J., F. Ziegler, …, M. Rief. 2011. The complex folding network of single calmodulin molecules. *Science.* 334:512–516.

12. Kaiser, C. M., D. H. Goldman, …, C. Bustamante. 2011. The ribosome modulates nascent protein folding. *Science.* 334:1723–1727.

13. Kruithof, M., and J. van Noort. 2009. Hidden Markov analysis of nucleosome unwrapping under force. *Biophys. J.* 96:3708–3715.

14. Stigler, J., and M. Rief. 2012. Hidden Markov analysis of trajectories in single-molecule experiments and the effects of missed events. *ChemPhysChem.* 13:1079–1086.

15. Liphardt, J., B. Onoa, …, C. Bustamante. 2001. Reversible unfolding of single RNA molecules by mechanical force. *Science.* 292:733–737.

16. Zhuang, X., and M. Rief. 2003. Single-molecule folding. *Curr. Opin. Struct. Biol.* 13:88–97.

17. Neupane, K., D. A. N. Foster, …, M. T. Woodside. 2016. Direct observation of transition paths during the folding of proteins and nucleic acids. *Science.* 352:239–242.

18. Žoldák, G., J. Stigler, …, M. Rief. 2013. Ultrafast folding kinetics and cooperativity of villin headpiece in single-molecule force spectroscopy. *Proc. Natl. Acad. Sci. USA.* 110:18156–18161.

19. Zhang, X., L. Ma, and Y. Zhang. 2013. High-resolution optical tweezers for single-molecule manipulation. *Yale J. Biol. Med.* 86:367–383.

20. Miller, J. A., S. J. Klippenstein, …, N. J. B. Green. 2009. Detailed balance in multiple-well chemical reactions. *Phys. Chem. Chem. Phys.* 11:1128–1137.

21. Alberty, R. A. 2004. Principle of detailed balance in kinetics. *J. Chem. Educ.* 81:1206–1209.

22. Baum, L. E., T. Petrie, …, N. Weiss. 1970. A maximization technique occurring in statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Stat.* 41:164–171.

23. Cecconi, C., E. A. Shank, …, S. Marqusee. 2005. Direct observation of the three-state folding of a single protein molecule. *Science.* 309:2057–2060.

24. Gao, Y., S. Zorman, …, Y. Zhang. 2012. Single reconstituted neuronal SNARE complexes zipper in three distinct stages. *Science.* 337:1340–1343.

25. Flomenbom, O., J. Klafter, and A. Szabo. 2005. What can one learn from two-state single-molecule trajectories? *Biophys. J.* 88:3780–3783.

26. Ma, L., Y. Kang, …, Y. Zhang. 2016. α-SNAP enhances SNARE zippering by stabilizing the SNARE four-helix bundle. *Cell Reports.* 15:531–539.

27. Jiao, J., A. A. Rebane, …, Y. Zhang. 2015. Kinetically coupled folding of a single HIV-1 glycoprotein 41 complex in viral membrane fusion and inhibition. *Proc. Natl. Acad. Sci. USA.* 112:E2855–E2864.

28. Woodside, M. T., P. C. Anthony, …, S. M. Block. 2006. Direct measurement of the full, sequence-dependent folding landscape of a nucleic acid. *Science.* 314:1001–1004.

29. Gebhardt, J. C. M., T. Bornschlögl, and M. Rief. 2010. Full distance-resolved folding energy landscape of one single protein molecule. *Proc. Natl. Acad. Sci. USA.* 107:2013–2018.

30. Azzalini, A. 2014. The Skew-Normal and Related Families. Cambridge University Press, New York.

31. Dudko, O. K., G. Hummer, and A. Szabo. 2006. Intrinsic rates and activation free energies from single-molecule pulling experiments. *Phys. Rev. Lett.* 96:108101.

32. Rebane, A. A., L. Ma, and Y. Zhang. 2016. Structure-based derivation of protein folding intermediates and energies from optical tweezers. *Biophys. J.* 110:441–454.

33. Ma, L., A. A. Rebane, …, Y. Zhang. 2015. Munc18-1-regulated stage-wise SNARE assembly underlying synaptic exocytosis. *eLife.* 4:e09580.

34. Moffitt, J. R., Y. R. Chemla, …, C. Bustamante. 2006. Differential detection of dual traps improves the spatial resolution of optical tweezers. *Proc. Natl. Acad. Sci. USA.* 103:9006–9011.

35. Noé, F. 2008. Probability distributions of molecular observables computed from Markov models. *J. Chem. Phys.* 128:244103.

36. Schwarz, G. 1978. Estimating dimension of a model. *Ann. Stat.* 6:461–464.

37. Yu, H., X. Liu, …, M. T. Woodside. 2012. Direct observation of multiple misfolding pathways in a single prion protein molecule. *Proc. Natl. Acad. Sci. USA.* 109:5283–5288.

38. Südhof, T. C., and J. E. Rothman. 2009. Membrane fusion: grappling with SNARE and SM proteins. *Science.* 323:474–477.

39. Zorman, S., A. A. Rebane, …, Y. Zhang. 2014. Common intermediates and kinetics, but different energetics, in the assembly of SNARE proteins. *eLife.* 3:e03348.

40. Sutton, R. B., D. Fasshauer, …, A. T. Brunger. 1998. Crystal structure of a SNARE complex involved in synaptic exocytosis at 2.4 Å resolution. *Nature.* 395:347–353.

41. Vivona, S., D. J. Cipriano, …, A. T. Brunger. 2013. Disassembly of all SNARE complexes by *n*-ethylmaleimide-sensitive factor (NSF) is initiated by a conserved 1:1 interaction between α-soluble NSF attachment protein (SNAP) and SNARE complex. *J. Biol. Chem.* 288:24984–24991.

42. Milescu, L. S., G. Akk, and F. Sachs. 2005. Maximum likelihood estimation of ion channel kinetics from macroscopic currents. *Biophys. J.* 88:2494–2515.

43. Kienker, P. 1989. Equivalence of aggregated Markov models of ion-channel gating. *Proc. Roy. Soc. B.* 236:269–309.

Supplemental Information

Hidden Markov Modeling with Detailed Balance and Its Application to Single Protein Folding

Yongli Zhang, Junyi Jiao, and Aleksander A. Rebane

Supplementary Text

Summary of calculating the log-likelihood and its derivative

We extended the method of Qin et al. to calculate the gradient of the likelihood (4, 5). Defining $b_i^{(t)} \equiv b_i\left(x^{(t)}\right), t = 1, \cdots, N_d$ with $x^{(t)}$ the t-th data point in the extension trajectory with a total number of data points N_d and

$$\boldsymbol{b}^{(t)} \equiv \begin{pmatrix} b_1^{(t)} & & 0 \\ & \ddots & \\ 0 & & b_N^{(t)} \end{pmatrix},$$ [S1]

we had

$$L = \boldsymbol{\pi}\boldsymbol{b}^{(1)} \bullet \boldsymbol{a}\boldsymbol{b}^{(2)} \bullet \cdots \bullet \boldsymbol{a}\boldsymbol{b}^{(N_T)} \bullet \mathbf{1},$$ [S2]

where $\boldsymbol{\pi} \equiv \left(\pi_1, \pi_2, \cdots \pi_N\right)$ with π_i the probability of the protein being in state i at time zero or $t = 1$ and $\mathbf{1}$ is an $N \times 1$ column vector containing all elements 1 with N the number of states in the model. The logarithm of the likelihood, or log-likelihood $\ln L$ can be computed using the forward-backward algorithm (1). As is shown in the following sections, the gradient of log-likelihood $\ln L$ relative to the energy g_{mn} with $m, n = 1, \cdots, N$ is

$$\frac{d \ln L}{d g_{mn}} = \sum_{i,j=1}^{N} D_{ij} \frac{\partial a_{ij}}{\partial g_{mn}},$$ [S3]

where

$$D_{ij} \equiv \frac{\partial \ln L}{\partial a_{ij}} = \frac{1}{L} \sum_{t=1}^{N_d - 1} \alpha_i^{(t)} \beta_j^{(t+1)} b_j^{(t+1)},$$ [S4]

and

$$\boldsymbol{\alpha}^{(t)} \equiv \boldsymbol{\pi}\boldsymbol{b}^{(1)} \bullet \boldsymbol{a}\boldsymbol{b}^{(2)} \bullet \cdots \bullet \boldsymbol{a}\boldsymbol{b}^{(t)}, t = 1, \cdots, N_d;$$

$$\boldsymbol{\beta}^{(t)} \equiv \boldsymbol{a}\boldsymbol{b}^{(t+1)} \bullet \cdots \bullet \boldsymbol{a}\boldsymbol{b}^{(N_T)} \bullet \mathbf{1}, t = 1, \cdots, N_d - 1;$$ [S5]

$$\boldsymbol{\beta}^{(N_T)} \equiv \mathbf{1}.$$

Note that $\boldsymbol{\alpha}^{(t)}$ and $\boldsymbol{\beta}^{(t)}$ are row and column vectors, respectively. Derivations of $\dfrac{\partial a_{ij}}{\partial g_{mn}}$ and gradients of $\ln L$ relative to other model parameters are shown in forthcoming sections.

Calculating the probability transition matrix a and its derivative

To facilitate computation of the transition probability matrix in Eq. 1, we perform a similarity transformation for the rate constant matrix $\boldsymbol{q}$,

$$\boldsymbol{Q} = \left(\boldsymbol{P}_{eq}\right)^{1/2} \boldsymbol{q} \left(\boldsymbol{P}_{eq}\right)^{-1/2}, \tag{S6}$$

where

$$\boldsymbol{P}_{eq} \equiv \begin{pmatrix} e^{-g_{11}} & & 0 \\ & \ddots & \\ 0 & & e^{-g_{NN}} \end{pmatrix} \tag{S7}$$

It can be shown that $\boldsymbol{Q}$ is a symmetric matrix with $\boldsymbol{Q}^T = \boldsymbol{Q}$. Using $\boldsymbol{Q}$, we can rewrite the transition probability matrix as

$$\boldsymbol{a} = \left(\boldsymbol{P}_{eq}\right)^{-1/2} \exp\left(\boldsymbol{Q}t\right)\left(\boldsymbol{P}_{eq}\right)^{1/2}. \tag{S8}$$

To calculate $\exp\left(\boldsymbol{Q}t\right)$, we express $\boldsymbol{Q}$ by eigenvalue decomposition (4), i.e.,

$$\boldsymbol{Q} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T = \sum_{i=1}^{N}\boldsymbol{U}_i\lambda_i \tag{S9}$$

, where

$$\boldsymbol{\Lambda} \equiv \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{pmatrix}, \tag{S10}$$

$$\boldsymbol{U} \equiv \left(\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_N\right), \tag{S11}$$

and

$$\boldsymbol{U}_i \equiv \boldsymbol{u}_i \boldsymbol{u}_i^T, \, i = 1, \cdots, N. \tag{S12}$$

Here $\lambda_1, \lambda_2, \cdots, \lambda_N$ are the eigenvalues of $\boldsymbol{Q}$ and $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_N$ are the corresponding orthonormal eigenvectors. Correspondingly,

$$\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}. \tag{S13}$$

It can be shown that

$$\sum_{i=1}^{N}\boldsymbol{U}_i = \boldsymbol{I}, \tag{S14}$$

and

$$\boldsymbol{U}_i\boldsymbol{U}_j = \delta_{ij}\boldsymbol{U}_i. \tag{S15}$$

Using Eqs. [S14] and [S15], we can calculate $\exp\left(\boldsymbol{Q}t\right)$ as follows:

$$\exp\left(\boldsymbol{Q}t\right) = \sum_{i=1}^{N}\boldsymbol{U}_i\exp\left(\lambda_i t\right). \tag{S16}$$

Correspondingly,

$$\boldsymbol{a} = \sum_{i=1}^{N}\left(\boldsymbol{P}_{eq}\right)^{-1/2}\boldsymbol{U}_i\left(\boldsymbol{P}_{eq}\right)^{1/2}\exp\left(\lambda_i t\right). \tag{S17}$$

This equation is used to compute the transition probability matrix, given the energy matrix.

We use gradient search to maximize the likelihood of observing a single molecule trajectory. This requires the derivatives of the transition probability matrix $\boldsymbol{a}$ with respective to

each element g_{mn} in the energy matrix $\boldsymbol{g}$. To this end, we calculated the derivative of $\boldsymbol{Q}$ and $\exp(\boldsymbol{Q}t)$, as is indicated by Eqs. [S8] and [S6]. We have

$$\frac{\partial q_{ij}}{\partial g_{mn}} = q_{mn}\delta_{im}\delta_{jm} + q_{nm}\delta_{in}\delta_{jn} - q_{mn}\delta_{im}\delta_{jn} - q_{nm}\delta_{in}\delta_{jm} + q_{mj}\delta_{mn}\delta_{im}, \qquad [S18]$$

where $\delta_{mn} = 1$ if $m = n$ and $\delta_{mn} = 0$ if $m \neq n$. Substituting Eq. [S18] into Eq. [S6], we have

$$\frac{\partial Q_{ij}}{\partial g_{mn}} = \begin{cases} q_{mn}\delta_{im}\delta_{jm} + q_{nm}\delta_{in}\delta_{jn} - Q_{mn}\delta_{im}\delta_{jn} - Q_{nm}\delta_{in}\delta_{jm}, & m \neq n, \\ \frac{1}{2}\left(Q_{im}\delta_{jm} + Q_{mj}\delta_{im}\right), & m = n. \end{cases} \qquad [S19]$$

Defining

$$\boldsymbol{X}(t) \equiv \frac{\partial \exp(\boldsymbol{Q}t)}{\partial g_{mn}}, \qquad [S20]$$

and then differentiating Eq. [S20] with respect to t (4), we have

$$\frac{d\boldsymbol{X}}{dt} = \boldsymbol{Q}\boldsymbol{X} + \frac{\partial \boldsymbol{Q}}{\partial g_{mn}}\exp(\boldsymbol{Q}t). \qquad [S21]$$

This differential equation on $\boldsymbol{X}$ can be solved with an initial condition $\boldsymbol{X}(0) = \boldsymbol{0}$, yielding

$$\boldsymbol{X}(t) = \int_0^t \exp(\boldsymbol{Q}(t-x))\frac{\partial \boldsymbol{Q}}{\partial g_{mn}}\exp(\boldsymbol{Q}x)\,dx. \qquad [S22]$$

Substituting Eqs. [S16] into Eq. [S22], we calculate the integral in Eq. [S22] as

$$\frac{\partial \exp(\boldsymbol{Q}t)}{\partial g_{mn}} = \sum_{i,j=1}^{N} \boldsymbol{U}_i \frac{\partial \boldsymbol{Q}}{\partial g_{mn}} \boldsymbol{U}_j \gamma_{ij}(t), \qquad [S23]$$

where

$$\gamma_{ij}(t) = \begin{cases} te^{\lambda_i t}, & \text{if } \lambda_i = \lambda_j \\ \dfrac{e^{\lambda_i t} - e^{\lambda_j t}}{\lambda_i - \lambda_j}, & \text{if } \lambda_i \neq \lambda_j \end{cases} \qquad [S24]$$

Note that as $t \to 0$,

$$\gamma_{ij}(t) \approx t. \qquad [S25]$$

Using Eq. [S19], we rewrite Eq. [S23] as

$$\left(\frac{\partial \exp(\boldsymbol{Q}t)}{\partial g_{mn}}\right)_{ij} = \sum_{k,l=1}^{N} \gamma_{kl}(t)\left[\begin{array}{l} q_{mn}(\boldsymbol{U}_k)_{im}(\boldsymbol{U}_l)_{jm} + q_{nm}(\boldsymbol{U}_k)_{in}(\boldsymbol{U}_l)_{jn} - \\ Q_{mn}(\boldsymbol{U}_k)_{im}(\boldsymbol{U}_l)_{jn} - Q_{nm}(\boldsymbol{U}_k)_{in}(\boldsymbol{U}_l)_{jm} \end{array}\right]$$
$$+ \frac{1}{2}\delta_{mn}\sum_{k,l=1}^{N}\gamma_{kl}(t)(\lambda_k + \lambda_l)(\boldsymbol{U}_k)_{im}(\boldsymbol{U}_l)_{jm} \qquad [S26]$$

Finally, substituting Eq. [S26] into Eq. [S8], we obtain the derivative of $\boldsymbol{a}$ relative to the energy parameter g_{mn}

$$\frac{\partial a_{ij}}{\partial g_{mn}} = \begin{cases} \exp\left[\frac{1}{2}\left(g_{ii} - g_{jj}\right)\right] \sum_{k,l=1}^{N} \gamma_{kl}(t) \left[\begin{array}{l} q_{mn}\left(U_k\right)_{im}\left(U_l\right)_{jm} + q_{nm}\left(U_k\right)_{in}\left(U_l\right)_{jn} - \\ Q_{mn}\left(U_k\right)_{im}\left(U_l\right)_{jn} - Q_{mn}\left(U_k\right)_{in}\left(U_l\right)_{jm} \end{array}\right], & m \neq n, \\ \frac{1}{2}\delta_{im}a_{mj} - \frac{1}{2}\delta_{jm}a_{im} + \frac{1}{2}\exp\left[\frac{1}{2}\left(g_{ii} - g_{jj}\right)\right] \sum_{k,l=1}^{N} \gamma_{kl}(t)\left(\lambda_k + \lambda_l\right)\left(U_k\right)_{im}\left(U_l\right)_{jm}, & m \neq n. \end{cases} \quad \text{[S27]}$$

Calculating the logarithm of likelihood $\ln L$

To calculate the likelihood L and its derivative using the forward-backward algorithm (1), we need to compute vectors $\boldsymbol{\alpha}^{(t)}$ and $\boldsymbol{\beta}^{(t)}$ defined in Eq. [S5]. Note that

$$L = \boldsymbol{\alpha}^{(t)} \cdot \boldsymbol{\beta}^{(t)}, \quad t = 1, \cdots, N_d. \quad \text{[S28]}$$

Both vectors can be recursively calculated:

$$\begin{cases} \boldsymbol{\alpha}^{(1)} = \boldsymbol{\pi b}^{(1)}, \\ \boldsymbol{\alpha}^{(t)} = \boldsymbol{\alpha}^{(t-1)}\boldsymbol{ab}^{(t)}, t = 2, \cdots, N_d; \\ \boldsymbol{\beta}^{(N_d)} = \mathbf{1}, \\ \boldsymbol{\beta}^{(t)} = \boldsymbol{ab}^{(t+1)}\boldsymbol{\beta}^{(t+1)}, t = 1, \cdots, N_d - 1. \end{cases} \quad \text{[S29]}$$

To overcome over-flow, normalization of both vectors is performed at each recursive step. For $\boldsymbol{\alpha}^{(t)}$, this can be computed as follows (1):

$$\begin{cases} c^{(1)} = \sum_{i=1}^{N} \alpha_i^{(1)}, \\ \hat{\boldsymbol{\alpha}}^{(1)} = \boldsymbol{\alpha}^{(1)} / c^{(1)}, \\ \overline{\boldsymbol{\alpha}}^{(t)} = \hat{\boldsymbol{\alpha}}^{(t-1)}\boldsymbol{ab}^{(t)}, \\ c^{(t)} = \sum_{i=1}^{N} \overline{\alpha}_i^{(t)}, \\ \hat{\boldsymbol{\alpha}}^{(t)} = \overline{\boldsymbol{\alpha}}^{(t)} / c^{(t)}, t = 2, \cdots, N_d. \end{cases} \quad \text{[S30]}$$

Correspondingly, $\hat{\boldsymbol{\beta}}^{(t)}$ is scaled as

$$\begin{cases} \hat{\boldsymbol{\beta}}^{(N_d)} = \boldsymbol{\beta}^{(N_d)} / c^{(N_d)}, \\ \overline{\boldsymbol{\beta}}^{(t)} = \boldsymbol{ab}^{(t+1)}\hat{\boldsymbol{\beta}}^{(t+1)}, \\ \hat{\boldsymbol{\beta}}^{(t)} = \overline{\boldsymbol{\beta}}^{(t)} / c^{(t)}, t = 1, \cdots, N_d - 1. \end{cases} \quad \text{[S31]}$$

It can be shown that

$$\boldsymbol{\alpha}^{(t)} = C^{(t)}\hat{\boldsymbol{\alpha}}^{(t)},$$
$$\boldsymbol{\beta}^{(t)} = E^{(t)}\hat{\boldsymbol{\beta}}^{(t)},$$

[S32]

where

$$C^{(t)} \equiv \prod_{s=1}^{t} c^{(s)},$$

$$E^{(t)} \equiv \prod_{s=t}^{N_d} c^{(s)}, \quad t = 1, \cdots, N_d,$$

[S33]

Using Eqs. [S32] and [S33], the likelihood L expressed in Eq. [S28] can be rewritten as

$$L = \boldsymbol{\alpha}^{(N_d)} \bullet \mathbf{1} = C^{(N_d)}\hat{\boldsymbol{\alpha}}^{(N_d)} \bullet \mathbf{1} = C^{(N_d)}.$$

[S34]

Thus,

$$\ln L = \sum_{t=1}^{N_d} \ln c^{(t)}.$$

[S35]

This equation is used to calculate the log-likelihood (1).

Computing the gradient of $\ln L$

To derive D_{ij} in Eq. [S4], we note that

$$
\begin{aligned}
\frac{\partial L}{\partial a_{ij}} &= \frac{\partial}{\partial a_{ij}}\left[\pi b^{(1)} \bullet ab^{(2)} \bullet \cdots \bullet ab^{(N_d)} \bullet \mathbf{1} \right] \\
&= \pi b^{(1)} \bullet \frac{\partial a}{\partial a_{ij}} b^{(2)} \bullet \cdots \bullet ab^{(N_d)} \bullet \mathbf{1} + \pi b^{(1)} a \bullet b^{(2)} \bullet \frac{\partial a}{\partial a_{ij}} b^{(3)} \bullet \cdots \bullet ab^{(N_d)} \bullet \mathbf{1} \\
&\quad + \cdots + \pi b^{(1)} a \bullet b^{(2)} \bullet \cdots \bullet ab^{(N_d-1)} \bullet \frac{\partial a}{\partial a_{ij}} b^{(N_d)} \bullet \mathbf{1} \\
&= \sum_{t=1}^{N_d-1} \boldsymbol{\alpha}^{(t)} \frac{\partial a}{\partial a_{ij}} b^{(t+1)} \boldsymbol{\beta}^{(t+1)} \\
&= \sum_{t=1}^{N_d-1} \sum_{k,l,m=1}^{N} \alpha_k^{(t)} \delta_{k,i} \delta_{l,j} b_{lm}^{(t+1)} \beta_m^{(t+1)} \\
&= \sum_{t=1}^{N_d-1} \alpha_i^{(t)} \beta_j^{(t+1)} b_j^{(t+1)}
\end{aligned}
$$

[S36]

From Eq. [S36], we have

$$D_{ij} = \frac{1}{L}\frac{\partial L}{\partial a_{ij}}$$

$$= \frac{1}{L}\sum_{t=1}^{N_d-1}\hat{\alpha}_i^{(t)}\hat{\beta}_j^{(t+1)}b_j^{(t+1)}C^{(t)}E^{(t+1)} \qquad [S37]$$

$$= \sum_{t=1}^{N_d-1}\hat{\alpha}_i^{(t)}\hat{\beta}_j^{(t+1)}b_j^{(t+1)}.$$

In deriving Eq. [S37], we have used the equation

$$C^{(t)}E^{(t+1)} = C^{(N_d)} = L, \qquad [S38]$$

and Eq. [S34]. Using Eq. [S37] and [S27], we obtain

$$\frac{d\ln L}{dg_{mn}} = \begin{cases} \sum_{i,j=1}^{N}D_{ij}\exp\left[\tfrac{1}{2}\left(g_{ii}-g_{jj}\right)\right]\sum_{k,l=1}^{N}\gamma_{kl}(t)\begin{bmatrix} q_{mn}\left(U_k\right)_{im}\left(U_l\right)_{jm}+q_{nm}\left(U_k\right)_{in}\left(U_l\right)_{jn} \\ -Q_{mn}\left(U_k\right)_{im}\left(U_l\right)_{jn}-Q_{mn}\left(U_k\right)_{in}\left(U_l\right)_{jm} \end{bmatrix}, & m\neq n, \\[2em] \tfrac{1}{2}\left\{\sum_{j=1}^{N}D_{mj}a_{mj}-\sum_{i=1}^{N}D_{im}a_{im}+\sum_{i,j=1}^{N}D_{ij}\exp\left[\tfrac{1}{2}\left(g_{ii}-g_{jj}\right)\right]\sum_{k,l=1}^{N}\gamma_{kl}(t)\left(\lambda_k+\lambda_l\right)\left(U_k\right)_{im}\left(U_l\right)_{jm}\right\}, & m=n. \end{cases}$$

$$[S39]$$

To simplify the expression in Eq. [S39], we first define the common term

$$H_{kl}^{(mn)} \equiv \gamma_{kl}(t)\sum_{i,j=1}^{N}D_{ij}\exp\left[\tfrac{1}{2}\left(g_{ii}-g_{jj}\right)\right]\left(U_k\right)_{im}\left(U_l\right)_{jn}. \qquad [S40]$$

Noting that

$$H_{kl}^{(mn)} = \gamma_{kl}(t)U_{mk}U_{nl}\sum_{i,j=1}^{N}D_{ij}\exp\left[\tfrac{1}{2}\left(g_{ii}-g_{jj}\right)\right]U_{ik}U_{jl}$$

$$= \gamma_{kl}(t)\left(U^T\right)_{km}U_{nl}\sum_{i,j=1}^{N}\left(U^T\right)_{ki}D_{ij}G_{ij}U_{jl} \qquad [S41]$$

where

$$G_{ij} \equiv \exp\left[\tfrac{1}{2}\left(g_{ii}-g_{jj}\right)\right], \qquad [S42]$$

we can write $H_{kl}^{(mn)}$ in a matrix form

$$\boldsymbol{H}^{(mn)} = \boldsymbol{\gamma}.*\left[\boldsymbol{U}^T(:,m)\bullet\boldsymbol{U}(n,:)\right].*\left[\boldsymbol{U}^T\bullet(\boldsymbol{D}.*\boldsymbol{G})\bullet\boldsymbol{U}\right]. \qquad [S43]$$

Here the .* denotes element-wise matrix multiplication, $\bullet$ represents the standard matrix multiplication, and $\boldsymbol{\gamma} = [\gamma_{kl}(t)]_{N\times N}$. Defining

$$\boldsymbol{h}^{(mn)} = \sum_{i,j=1}^{N}\boldsymbol{H}_{ij}^{(mn)}, \qquad [S44]$$

we can rewrite Eq. [S39] as

$$\frac{d \ln L}{dg_{mn}} = \begin{cases} q_{mn} h^{(mm)} + q_{nm} h^{(nn)} - Q_{mn}\left(h^{(mn)} + h^{(nm)}\right), & m < n, \\[2em] \frac{1}{2}\left\{\displaystyle\sum_{j=1}^{N} D_{mj} a_{mj} - \sum_{i=1}^{N} D_{im} a_{im} + \sum_{k,l=1}^{N}\left(\lambda_k + \lambda_l\right) H_{kl}^{(mm)}\right\}, & m = n. \end{cases}$$ [S45]

Using Eq. [S2], we calculate the gradient of $\ln L$ with respect to μ_i

$$\frac{\partial \ln L}{\partial \mu_i} = \frac{1}{L}\frac{\partial}{\partial \mu_i}\left[\boldsymbol{\pi b}^{(1)} \boldsymbol{\cdot} \boldsymbol{ab}^{(2)} \boldsymbol{\cdot} \cdots \boldsymbol{\cdot} \boldsymbol{ab}^{(N_d)} \boldsymbol{\cdot 1}\right]$$

$$= \frac{1}{L}\left[\boldsymbol{\pi} \frac{\partial \boldsymbol{b}^{(1)}}{\partial \mu_i}\boldsymbol{\beta}^{(1)} + \sum_{t=1}^{N_d-1}\boldsymbol{\alpha}^{(t)} \boldsymbol{\cdot a} \frac{\partial \boldsymbol{b}^{(t+1)}}{\partial \mu_i}\boldsymbol{\cdot \beta}^{(t+1)}\right]$$

$$= \frac{1}{L}\left[\sum_{k,m=1}^{N}\pi_k \frac{\partial b_{km}^{(1)}}{\partial \mu_i}\beta_m^{(1)} + \sum_{t=1}^{N_d-1}\sum_{l,k,m=1}^{N}\alpha_l^{(t)} a_{lk}\frac{\partial b_{km}^{(t+1)}}{\partial \mu_i}\beta_m^{(t+1)}\right]$$

$$= \frac{1}{L}\left[\sum_{k,m=1}^{N}\pi_k \delta_{km}\delta_{im}\frac{x^{(1)}-\mu_i}{\sigma_i^2}b_i^{(1)}\beta_m^{(1)} + \sum_{t=1}^{N_d-1}\sum_{l,k,m=1}^{N}\alpha_l^{(t)} a_{lk}\delta_{km}\delta_{im}\frac{x^{(t+1)}-\mu_i}{\sigma_i^2}b_i^{(t+1)}\beta_m^{(t+1)}\right]$$

$$= \frac{1}{L}\left[\pi_i \frac{x^{(1)}-\mu_i}{\sigma_i^2}b_i^{(1)}\beta_i^{(1)} + \sum_{t=1}^{N_d-1}\sum_{k=1}^{N}\alpha_k^{(t)} a_{ki}\frac{x^{(t+1)}-\mu_i}{\sigma_i^2}b_i^{(t+1)}\beta_i^{(t+1)}\right]$$

$$= \frac{1}{L\sigma_i^2}\left[\pi_i b_i^{(1)}\left(x^{(1)}-\mu_i\right)\beta_i^{(1)} + \sum_{t=1}^{N_d-1}\alpha_i^{(t+1)}\left(x^{(t+1)}-\mu_i\right)\beta_i^{(t+1)}\right]$$

$$= \frac{1}{\sigma_i^2}\sum_{t=1}^{N_d}\frac{\alpha_i^{(t)}\beta_i^{(t)}}{L}\left(x^{(t)}-\mu_i\right).$$ [S46]

In deriving Eq. [S46], we have used

$$\frac{\partial b_{km}^{(t)}}{\partial \mu_i} = \delta_{km}\delta_{im}\frac{x^{(t)}-\mu_i}{\sigma_i^2}b_i^{(t)}.$$ [S47]

Defining

$$\chi_i^{(t)} \equiv \frac{\hat{\alpha}_i^{(t)}\hat{\beta}_i^{(t)}}{L}, \ i = 1,\cdots,N,$$ [S48]

and noting Eqs. [S28] and [S32], we have

$$\chi_i^{(t)} = \frac{\hat{\alpha}_i^{(t)}\hat{\beta}_i^{(t)}}{\boldsymbol{\alpha}^{(t)} \boldsymbol{\cdot \beta}^{(t)}} = \frac{\hat{\alpha}_i^{(t)}\hat{\beta}_i^{(t)}}{\displaystyle\sum_{j=1}^{N}\hat{\alpha}_j^{(t)}\hat{\beta}_j^{(t)}}.$$ [S49]

Using Eq. [S49], we can rewrite Eq. [S46] as

$$\frac{\partial \ln L}{\partial \mu_i} = \frac{1}{\sigma_i^2} \sum_{t=1}^{N_d} \chi_i^{(t)} \left(x^{(t)} - \mu_i \right).$$ [S50]

The gradients of $\ln L$ with respect to other parameters can be similarly derived (4):

$$\frac{\partial \ln L}{\partial \sigma_i^2} = \frac{1}{2\sigma_i^2} \sum_{t=1}^{N_d} \chi_i^{(t)} \left[-1 + \frac{\left(x^{(t)} - \mu_i \right)^2}{\sigma_i^2} \right],$$

$$\frac{\partial \ln L}{\partial \pi_i} = \hat{\beta}_i^{(1)} b_i^{(1)}.$$ [S51]

Derivations related to the skew normal distribution

To calculate the mean of the skew normal distribution shown in Eq. 5, we first define

$$u = \frac{x - \mu}{\sqrt{2}\sigma}.$$ [S52]

Then

$$\begin{aligned}
\overline{x} &= \int_{-\infty}^{+\infty} x f(x) \, dx \\
&= \frac{1}{\pi\sigma} \int_{-\infty}^{+\infty} x e^{-\frac{(x-\mu)^2}{2\sigma^2}} \int_{-\infty}^{\zeta\left(\frac{x-\mu}{\sigma}\right)} e^{-\frac{t^2}{2}} \, dt \, dx \\
&= \frac{2}{\pi} \int_{-\infty}^{+\infty} \left(\mu + \sqrt{2}\sigma u \right) e^{-u^2} \int_{-\infty}^{\zeta u} e^{-t^2} \, dt \, du \\
&= \frac{2\mu}{\pi} \int_{-\infty}^{+\infty} e^{-u^2} \int_{-\infty}^{\zeta u} e^{-t^2} \, dt \, du + \frac{2\sqrt{2}\sigma}{\pi} \int_{-\infty}^{+\infty} u e^{-u^2} \int_{-\infty}^{\zeta u} e^{-t^2} \, dt \, du.
\end{aligned}$$ [S53]

Here we have changed the variables according to Eq. [S52] and $t \to \sqrt{2}t$. We further change the variables (u,t) to the polar coordinate (r,θ), then

$$\overline{x} = \frac{2\mu}{\pi} \int_0^{+\infty} r e^{-r^2} \, dr \int_{\theta_1}^{\theta_2} d\theta + \frac{2\sqrt{2}\sigma}{\pi} \int_0^{+\infty} r^2 e^{-r^2} \, dr \int_{\theta_1}^{\theta_2} \cos\theta \, d\theta,$$ [S54]

where the integration is over the area below the line $t = \zeta u$ with $\theta_1 = -\pi + \theta_2$ and $\tan\theta_2 = \zeta$.
Using the following formulae for Gaussian integrals

$$\int_0^{+\infty} x^{2n} e^{-\frac{x^2}{\alpha^2}} dx = \sqrt{\pi} \frac{\alpha^{2n+1}(2n-1)!!}{2^{n+1}},$$

$$\int_0^{+\infty} x^{2n+1} e^{-\frac{x^2}{\alpha^2}} dx = \frac{n!}{2}\alpha^{2n+2}, \quad n = 1, 2, \cdots \tag{S55}$$

we can simplify Eq. [S54] as

$$\bar{x} = \mu + \sqrt{\frac{2}{\pi}}\sin\theta_2$$

$$= \mu + \sigma\delta\sqrt{\frac{2}{\pi}}, \tag{S56}$$

where

$$\delta \equiv \frac{\zeta}{\sqrt{1+\zeta^2}}. \tag{S57}$$

The standard deviation and skewness of the skew normal distribution shown in Eqs. 8 and 9 can be similarly derived. These formulae provide a convenient method to fit the skew normal distribution to the PDF of an extension trajectory $x^{(t)}, t = 1, \cdots, N_d$. We first compute its average $\bar{x}$, standard deviation ω, and skewness γ. We then solve Eq. 9 for the parameter δ, i.e.,

$$|\delta| = \sqrt{\frac{\pi}{2} \frac{|\gamma|^{\frac{2}{3}}}{|\gamma|^{\frac{2}{3}} + \left[(4-\pi)/2\right]^{\frac{2}{3}}}}, \tag{S58}$$

where the sign of δ is the same as the sign of γ. The shape parameter can be obtained by

$$\zeta \equiv \frac{\delta}{\sqrt{1-\delta^2}}. \tag{S59}$$

Finally, the location μ and the scaling factor σ can be derived from Eqs. 7 and 8. The parameters obtained by this method were close to those estimated by the maximum likelihood approach.

Then we calculate the derivative of the skew normal distribution with respect to its parameters. Based on the expressions in Eqs. 5 and 6, we have the probability density to observe an extension x is

$$b = \frac{1}{\pi\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \int_{-\infty}^{\zeta\left(\frac{x-\mu}{\sigma}\right)} e^{-\frac{s^2}{2}} ds. \tag{S60}$$

Then

$$\frac{\partial b}{\partial \mu} = \frac{1}{\pi\sigma}\frac{(x-\mu)}{\sigma^2}e^{-\frac{(x-\mu)^2}{2\sigma^2}}\int_{-\infty}^{\varsigma\left(\frac{x-\mu}{\sigma}\right)}e^{-\frac{s^2}{2}}ds - \frac{1}{\pi\sigma}\frac{\varsigma}{\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}e^{-\frac{1}{2}\varsigma^2\left(\frac{x-\mu}{\sigma}\right)^2}$$

$$= \frac{b}{\sigma^2}\left\{x-\mu-\varsigma\sigma e^{-\frac{1}{2}\varsigma^2\left(\frac{x-\mu}{\sigma}\right)^2}\left[\int_{-\infty}^{\varsigma\left(\frac{x-\mu}{\sigma}\right)}e^{-\frac{s^2}{2}}ds\right]^{-1}\right\} \tag{61}$$

$$= \frac{\sqrt{2}b}{\sigma}\left[u-\varsigma\Psi(\varsigma u)\right],$$

where u has been defined by Eq. [S52] and

$$\Psi(x) = \frac{e^{-x^2}}{\sqrt{2}\int_{-\infty}^{\sqrt{2}x}e^{-\frac{s^2}{2}}ds} = \frac{1}{\sqrt{\pi}}\frac{e^{-x^2}}{1+\mathrm{erf}(x)}. \tag{S62}$$

Denoting

$$u_i^{(t)} = \frac{x^{(t)}-\mu_i}{\sqrt{2}\sigma_i},$$

$$\Psi_i^{(t)} = \Psi\left(\varsigma_i u_i^{(t)}\right), \tag{S63}$$

and using Eq. (61), we have

$$\frac{\partial \ln L}{\partial \mu_i} = \frac{\sqrt{2}}{\sigma_i}\sum_{t=1}^{N_d}\chi_i^{(t)}\left(u_i^{(t)}-\varsigma_i\Psi_i^{(t)}\right), \quad i=1,\cdots,N. \tag{S64}$$

Similarly, we derive

$$\frac{\partial \ln L}{\partial \sigma_i^2} = \frac{1}{2\sigma_i^2}\sum_{t=1}^{N_T}\chi_i^{(t)}\left(-1+2\left(u_i^{(t)}\right)^2-2\varsigma_i u_i^{(t)}\Psi_i^{(t)}\right),$$

$$\frac{\partial \ln L}{\partial \varsigma_i} = 2\sum_{t=1}^{N_T}\chi_i^{(t)}u_i^{(t)}\Psi_i^{(t)}. \tag{S65}$$

Equations [S64] and [S65] return to Eqs. [S50] and [S51] when $\varsigma_i = 0, i = 1,\cdots,N$. Equations [S64] and [S65] are used to compute the gradient of the log-likelihood with respect to the parameters in the skew normal distribution. Note that the function $\Psi(x)$ defined in Eq. [S62] monotonically decreases as x increases and $\Psi(x) \to 0$ as $x \to +\infty$. However, the computation of $\Psi(x)$ tends to overflow as $x < -5$ because of MATLAB's lower precision to compute the

error function than to calculate the exponential function in Eq. [S62]. To avoid the overflow, we make use of the asymptotic expansion of the error function for large x, i.e.,

$$\text{erf}(x) \sim 1 - \frac{e^{-x^2}}{\sqrt{\pi}} \left(x^{-1} - \frac{1}{2}x^{-3} + \frac{3}{4}x^{-5} - \frac{15}{8}x^{-7} \right). \tag{S66}$$

Substituting Eq. [S66] into Eq. [S62], we have

$$\Psi(x) \approx -\frac{1}{x^{-1}\left(1 - \frac{1}{2}x^{-2} + \frac{3}{4}x^{-4} - \frac{15}{8}x^{-6}\right)} \tag{S67}$$

for $x < -5$.

Baseline correction

We represent the discrete baseline by a series of piecewise-linear functions (9)

$$h^{(t)} = \sum_{r=1}^{R} \kappa_r h_r^{(t)}, \ \text{t} = 1, \cdots, \text{N}_T \tag{S68}$$

where the r-piece linear function

$$h_r^{(t)} = \begin{cases} 1 - \dfrac{|t - rM|}{M}, & (r-1)M < t < (r+1)M, \\ 0, & \text{others}, \end{cases} \quad r = 1, \cdots, R. \tag{S69}$$

and κ_r is its coefficient to be determined. Here $M = ceil\left[N_T/(R-1)\right]$ is the number of data points in each piece of data, where $ceil$ is a MATLAB function that rounds a real number to the nearest integer greater than or equal to that number. Using Eq. [S69], Eq. [S68] can be simplified as

$$h^{(t)} = \kappa_m + (\kappa_{m+1} - \kappa_m)\frac{l}{M}, \quad m = ceil(t/M), l = \text{mod}(t, M), \tag{S70}$$

where mod is the modulus function.

To compute the log-likelihood and its gradient relative to the parameters κ_r, we replace the extension data $x^{(t)}$ at data point t with

$$x^{(t)'} = x^{(t)} - h^{(t)} = x^{(t)} - \sum_{r=1}^{R} \kappa_r h_r^{(t)}. \tag{S71}$$

The gradient can be calculated as follows assuming a normal noise distribution:

$$\frac{\partial \ln L}{\partial \kappa_r} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \sum_{t=(r-1)+1}^{(r+1)M-1} \chi_i^{(t)} h_r^{(t)} \left(x^{(t)'} - \mu_i \right). \tag{S72}$$

Note that the baseline coefficients κ_r, $r = 1, \cdots, R$ and state extensions μ_i, $i = 1, \cdots, N$ are not independent. To remove the dependence, we fixed the extension of state 1 or μ_1 to an approximate extension of state 1 or zero and optimized other parameters. Thus, the baseline essentially served as a time-dependent extension of state 1 and HMM yielded the extensions of other states relative to the extension of state 1.

Computation flow chart

1. Calculate rate constant matrices Q and q

$$Q_{ij} = \begin{cases} k_m \exp\left(\dfrac{g_{ii} + g_{jj}}{2} - g_{ij} \right), & i \neq j; \\[2ex] -k_m e^{g_{ii}} \displaystyle\sum_{j \neq i}^{N} e^{-g_{ij}}, & i = j. \end{cases} \tag{S73}$$

$$q_{ij} = \exp\left[\tfrac{1}{2}\left(g_{ii} - g_{jj} \right) \right] Q_{ij}. \tag{S74}$$

2. Compute eigenvalues λ_i, $i = 1, \cdots, N$ and eigenvectors u_i of Q and associated quantities.

$$Q = U \Lambda U^T = \sum_{i=1}^{N} U_i \lambda_i$$

$$U = \left(u_1, u_2, \cdots, u_N \right)$$

$$U_i = u_i u_i^T. \tag{S75}$$

$$\gamma_{ij}(t) = \begin{cases} t e^{\lambda_i t}, & \text{if } \lambda_i = \lambda_i \\[2ex] \dfrac{e^{\lambda_i t} - e^{\lambda_j t}}{\lambda_i - \lambda_j}, & \text{if } \lambda_i \neq \lambda_i. \end{cases}$$

3. Calculate transition probability matrix a:

$$a_{ij} = \exp\left[\tfrac{1}{2}\left(g_{ii} - g_{jj} \right) \right] \sum_{k=1}^{N} (U_k)_{ij} \exp(\lambda_k t) \tag{S76}$$

4. Calculate the matrices $H^{(mn)}$ and $h^{(mn)}$

$$H^{(mn)} = \gamma .* \left[U^T(:,m) \bullet U(n,:) \right] .* \left[U^T \bullet (D.*G) \bullet U \right],$$

$$h^{(mn)} = \sum_{i,j=1}^{N} H_{ij}^{(mn)}. \tag{S77}$$

5. Calculate the log-likelihood $\ln L$.
6. Calculate the gradient of $\ln L$ with respect to the energy matrix

$$\frac{\partial \ln L}{\partial g_{mn}} = \begin{cases} q_{mn} h^{(mm)} + q_{nm} h^{(nn)} - Q_{mn}\left(h^{(mn)} + h^{(nm)} \right), & m < n, \\[3ex] \tfrac{1}{2}\left\{ \displaystyle\sum_{j=1}^{N} D_{mj} a_{mj} - \sum_{i=1}^{N} D_{im} a_{im} + \sum_{k,l=1}^{N} \left(\lambda_k + \lambda_l \right) H_{kl}^{(mm)} \right\}, & m = n. \end{cases} \tag{S78}$$

7. Optimize model parameters using MATLAB function *fminunc* or *fmincon*.

Supplementary Tables

TABLE S1 Activation energy matrix corresponding to the energy matrix in Table 1.

State #	1	2	3	4	5	6	7
1	0	7	9	8	9	8	9
2	7	0	9	10	7	9	9
3	7	7	0	9	9	9	9
4	8	10	11	0	8	9	10
5	8	6	10	7	0	8	9
6	9	10	12	10	10	0	9
7	9	9	11	10	10	8	0

TABLE S2 Number of transitions in one of the simulated extension trajectories with 10 s total length of time.

State #	1	2	3	4	5	6	7
1	4977	908	123	360	190	426	148
2	908	4687	131	151	664	205	146
3	123	131	561	22	21	30	17
4	360	151	22	5592	283	174	66
5	190	664	21	283	1054	130	62
6	426	205	30	174	130	17949	410
7	148	146	17	66	62	410	5798

TABLE S3 Comparison of state populations or probabilities derived from the extension trajectories using HMM-DB and HMM-EM. The extension trajectories were simulated using the same model parameters (called inputs, see Table 1 for energy parameters) but for two lengths of time T_d, T_d =10 s for one and T_d =60 s for the other.

Method	Time (s)	1	2	3	4	5	6	7
Input	-	0.138	0.138	0.019	0.138	0.051	0.376	0.138
HMM-DB	10	0.144	0.137	0.017	0.134	0.047	0.386	0.133
HMM-EM	10	0.144	0.138	0.017	0.134	0.047	0.387	0.133

HMM-DB	60	0.139	0.140	0.019	0.137	0.051	0.375	0.139
HMM-EM	60	0.139	0.140	0.019	0.137	0.051	0.375	0.138

TABLE S4   Comparison of the state lifetimes derived from HMM-DB and HMM-EM and calculated from the model parameters.

Method	*Time (s)*	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Input	-	0.512	0.447	0.432	1.109	0.236	2.570	1.450
HMM-DB	10	0.518	0.442	0.402	1.093	0.224	2.561	1.431
HMM-EM	10	**0.660**	**0.620**	**0.516**	**1.243**	**0.356**	**2.741**	**1.552**
HMM-DB	60	0.507	0.446	0.424	1.091	0.239	2.568	1.426
HMM-EM	60	0.506	0.446	0.426	1.091	0.239	2.569	1.427

TABLE S5   Comparison of the best-fit parameters derived from the simulated extension trajectories by HMM-DB and HMM-EM and the input model parameters. The 10 fitting parameters in the transition state energy that deviate most from their respective input parameters are highlighted in bold.

Parameter #	1	2	3	4	5	6	7	8	9	10	11
Parameter	g_1	g_2	g_3	g_4	g_5	g_6	g_{12}	g_{13}	g_{14}	g_{15}	g_{16}
Input	0	0	2	0	1	-1	7	9	8	9	8
DB	-0.08	-0.03	2.04	-0.01	1.04	-1.06	6.93	8.80	7.99	9.13	7.86
DB Error	0.02	0.02	0.06	0.03	0.03	0.03	0.04	0.09	0.04	0.15	0.04
EM	-0.07	-0.03	2.04	-0.01	1.04	-1.06	**7.28**	9.16	**8.20**	9.04	8.01
DB	-0.01	-0.01	2.01	0.01	0.99	-1.00	6.97	9.02	7.98	9.03	8.01
DB Error	0.01	0.01	0.02	0.01	0.01	0.01	0.02	0.04	0.02	0.05	0.02
EM	0.00	-0.01	2.01	0.01	0.99	-1.00	6.97	9.03	7.98	9.05	8.01

12	13	14	15	16	17	18	19	20	21	22
g_{17}	g_{23}	g_{24}	g_{25}	g_{26}	g_{27}	g_{34}	g_{35}	g_{36}	g_{37}	g_{45}
9	9	10	7	9	9	11	11	11	11	8
9.03	9.06	9.86	6.97	9.01	9.03	10.93	**11.42**	10.90	**11.54**	7.99
0.08	0.13	0.23	0.04	0.09	0.08	0.34	0.66	0.26	0.38	0.06
9.12	**9.39**	**9.02**	**7.59**	**8.78**	9.11	10.90	**11.32**	10.83	**11.70**	**8.46**
8.95	9.00	10.16	7.00	8.95	9.02	10.67	11.26	10.98	10.90	7.99
0.03	0.05	0.12	0.01	0.04	0.03	0.11	0.22	0.11	0.09	0.02
8.95	9.00	10.11	6.99	8.94	9.01	10.73	11.24	10.99	10.91	7.99

23	24	25	26	27	28	29	30	31	32	33
g_{46}	g_{47}	g_{56}	g_{57}	g_{67}	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6
9	10	9	10	8	0	10	20	30	40	50
8.97	9.97	8.97	9.83	7.96	0.12	10.04	19.69	30.00	40.00	50.03
0.07	0.12	0.10	0.14	0.04	0.06	0.06	0.18	0.05	0.11	0.03
8.92	9.93	**9.24**	10.03	8.05	0.11	10.03	19.69	30.01	40.01	50.03
9.04	9.98	9.05	9.97	7.99	0.04	9.99	20.02	30.00	40.02	49.99
0.03	0.05	0.04	0.06	0.02	0.02	0.02	0.07	0.02	0.04	0.01
9.03	9.98	9.05	9.98	7.99	0.02	9.97	20.03	30.00	40.02	49.99

34	35	36	37	38	39	40	41
μ_7	σ_1^2	σ_2^2	σ_3^2	σ_4^2	σ_5^2	σ_6^2	σ_7^2
60	16	16	16	16	16	16	16
59.96	16.38	16.48	13.71	16.00	15.86	16.00	16.23
0.05	0.33	0.40	1.01	0.32	0.67	0.18	0.31
59.96	16.32	16.61	13.89	16.01	15.80	16.02	16.24
60.01	16.22	16.11	15.07	15.84	16.43	16.13	15.90
0.02	0.13	0.15	0.43	0.13	0.26	0.07	0.12
60.01	16.13	16.23	15.08	15.86	16.42	16.15	15.92

TABLE S6 Comparison of the best-fit parameters derived from the SNARE zippering data by HMM under different conditions: HMM-DB with the skew normal noise distribution (DB-SN) and the normal noise distribution (DB-N and DB-N-k_m) and by HMM-EM with the normal noise distribution (EM-N). All HMM-DB used a molecular transition rate constant $k_m=1\times10^{-6}$ s^{-1}, except that in DB-N-k_m, $k_m=1\times10^{-5}$ s^{-1}. Comparing the results from DB-N and DB-N-k_m, the k_m decrease only reduced the energy of the transition states (bold numbers) by an approximate amount of ln10=2.30 predicted by Eq. 17. As a result, the kinetics of SNARE zippering derived by both HMM methods were identical and thus independent of k_m. For DB-SN, the average state extensions ($\bar{x}$) and their variance (ω^2) were calculated using Eqs. 7 and 8 and found to be close to the corresponding parameters directly derived from DB-N and EM. The fitting errors for DB-SN are shown in Fig. S2.

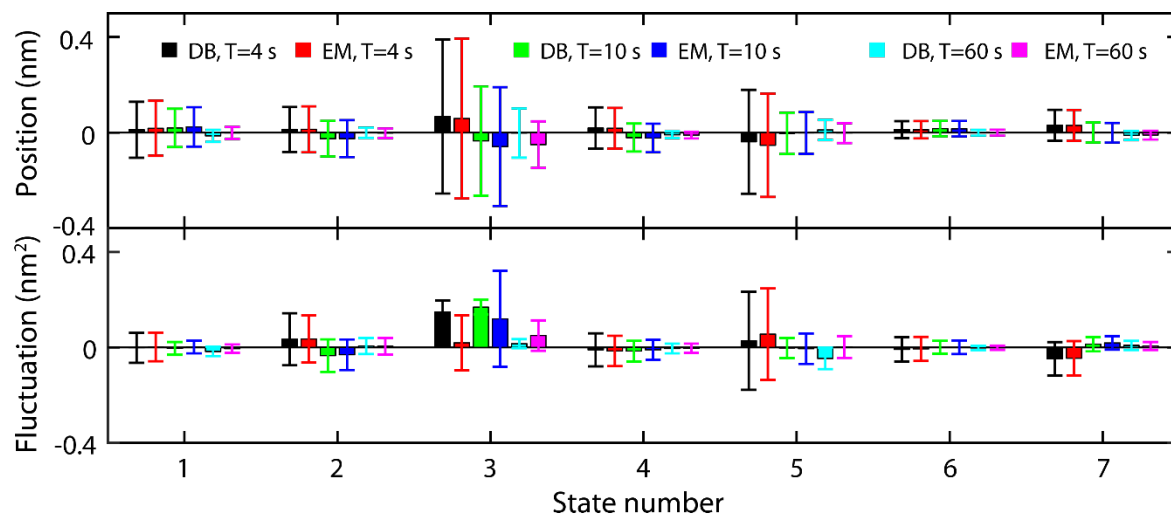| Parameter #	1	2	3	4	5	6	7	8	9	10	11
Parameter						μ_1	μ_2	μ_3	σ_1^2	σ_2^2	σ_3^2
	g_1	g_2	g_{12}	g_{13}	g_{23}	v_1	v_2	v_3	ω_1^2	ω_2^2	ω_3^2
DB-N	-0.94	-1.16	7.38	20.5	9.20	735.95	742.33	753.88	2.97	5.37	3.95
DB-N-k_m	-0.94	-1.16	**5.08**	**19.4**	**6.89**	735.95	742.33	753.88	2.97	5.37	3.95
DB-SN	-0.93	-1.16	7.38	22.1	9.20	735.95	742.32	753.87	2.95	5.34	4.02
EM-N	-0.94	-1.16	7.39	21.5	9.20	735.95	742.33	753.88	2.98	5.39	3.97

Supplementary Figures

FIGURE S1 Comparison of the average deviations and their standard deviations of the best-fit state positions (top panel) and fluctuations (bottom panel) from HMM-DB and HMM-EM of ten simulated trajectories for each time lengths of 4 s, 10 s, and 60 s.
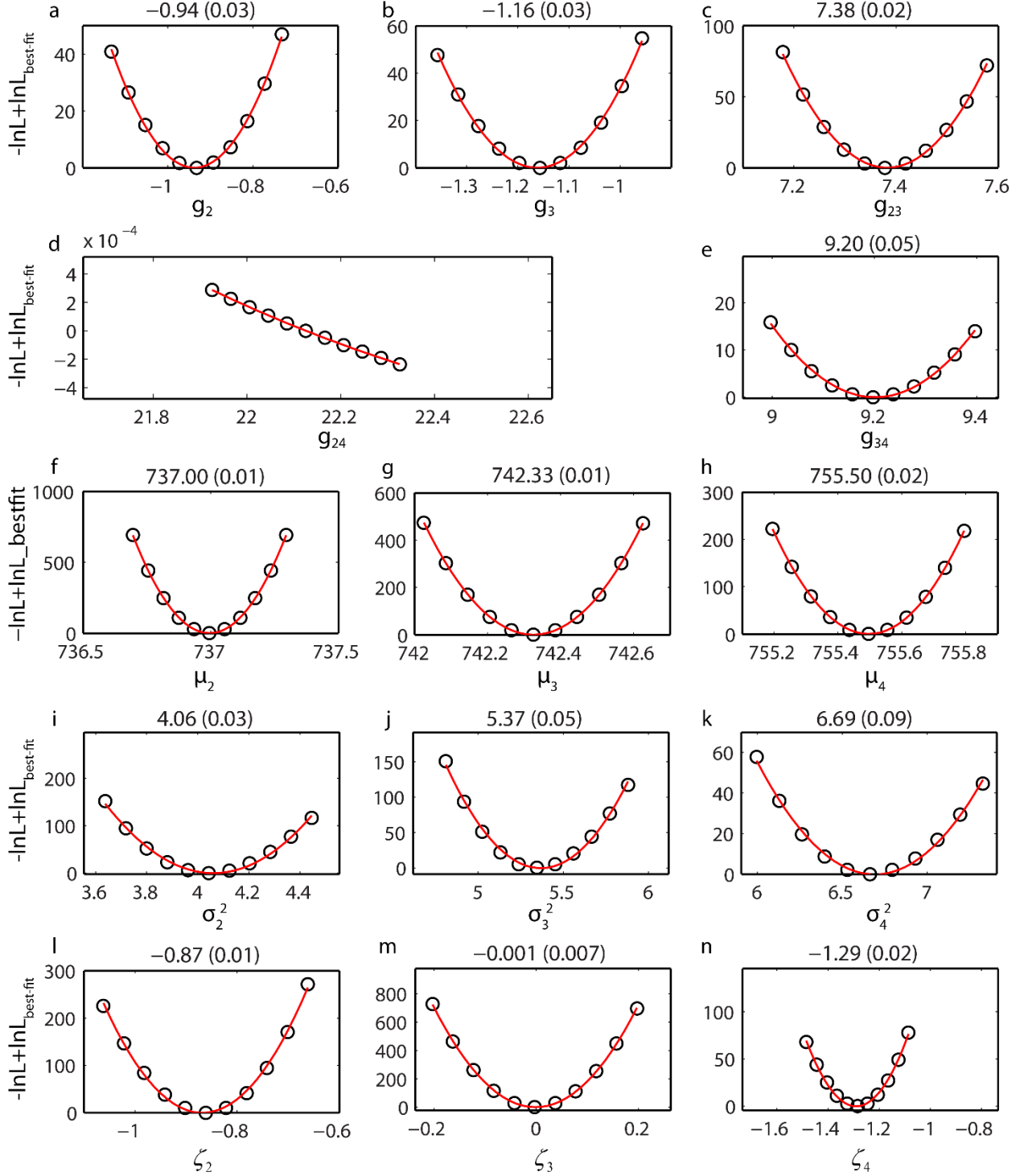
Figure S2 Change of the log-likelihood as a function of the change of each fitting parameter around its best-fit value. The measured single molecule trajectory of SNARE zippering in the absence of SNAP (Fig. 6A) was analyzed using HMM-DB with the skew normal noise distribution. The model parameters consisted of energy matrix elements $g_{ij}, i \leq j, i, j = 2, 3, 4$, state locations μ_i, fluctuations σ_i^2, and shapes ζ_i for $i = 2, 3, 4$. Varying each model parameter

around its best-fit value decreases the likelihood L relative to the maximum likelihood $L_{best-fit}$. Note that the y-axes of all subfigures show the calculated deviation $y = -\ln L + \ln L_{best-fit}$ (symbols). The deviation was fitted with a quadratic function $y = \frac{1}{2\xi^2}(x - x_0)^2 + y_0$ (red curve) to obtain the optimal parameter value x_0 and the standard deviation of the derived optimal value ξ that is called fitting error in the text. The optimal value x_0 is generally close to the corresponding best-fit value from HMM and is shown above each plot followed by its standard deviation ξ in parenthesis. The likelihood became insensitive to the energy barrier g_{24} between the zippered state 2 and the unzipped state as g_{24} exceeded 21 k_BT, indicating negligible transitions between the two states.
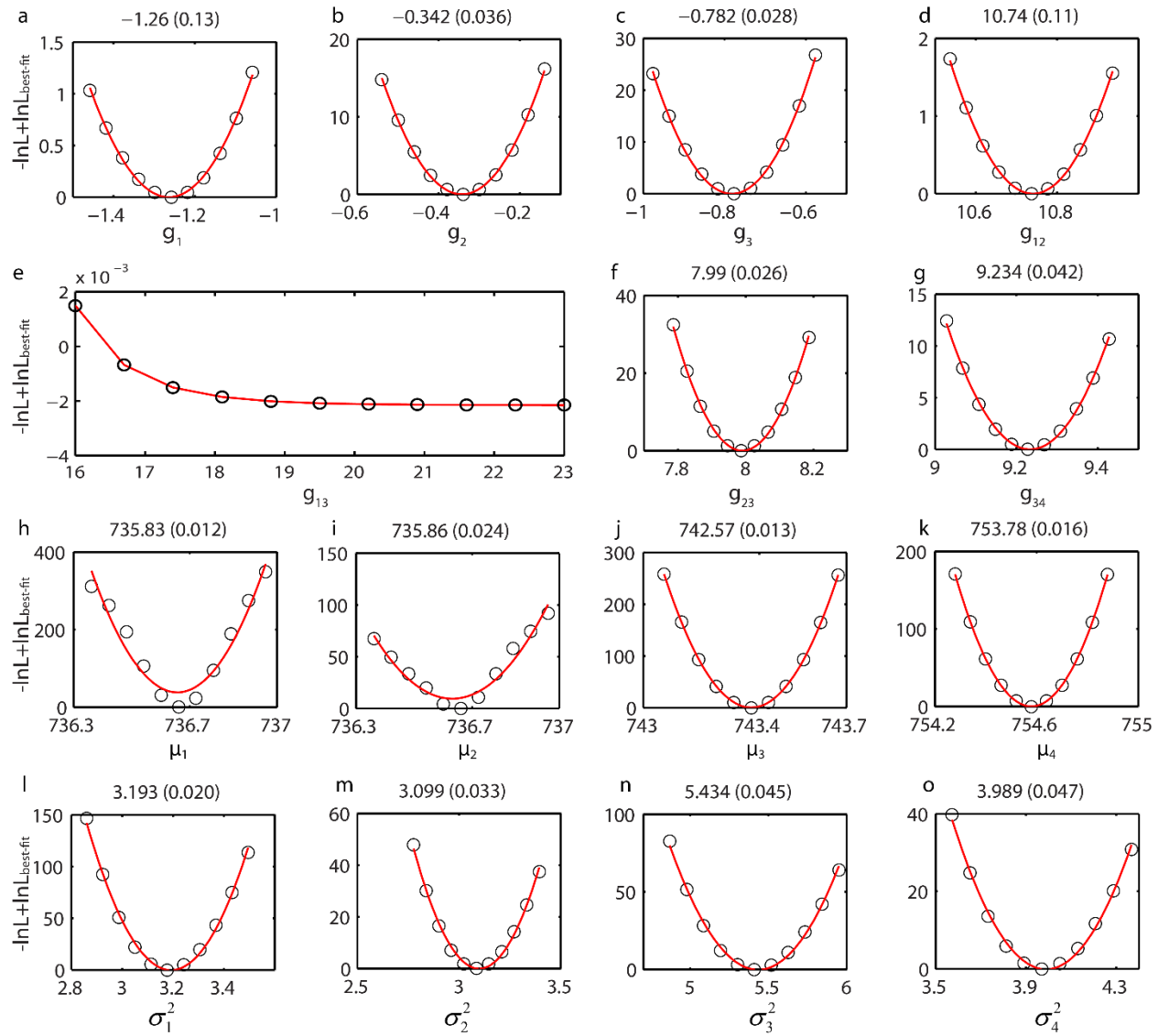

FIGURE S3 Change of the log-likelihood as a function of the change of each fitting parameter around its best-fit value. The measured single molecule trajectory of SNARE zippering in the

presence of SNAP (Fig. 6A) was analyzed using HMM-DB with a normal noise distribution. The likelihood became insensitive to the energy barrier g_{13} between the SNAP bound state 1 and the partially zippered state as g_{13} exceeded 16 k_BT, indicating negligible transitions between the two states. The direct transition between state 2 and 4 was inhibited by setting $g_{24} = 25$ during the maximum-likelihood estimation. In addition, the average positions of states 1 and 2 were constrained to be equal, i.e., $\mu_1 = \mu_2$.
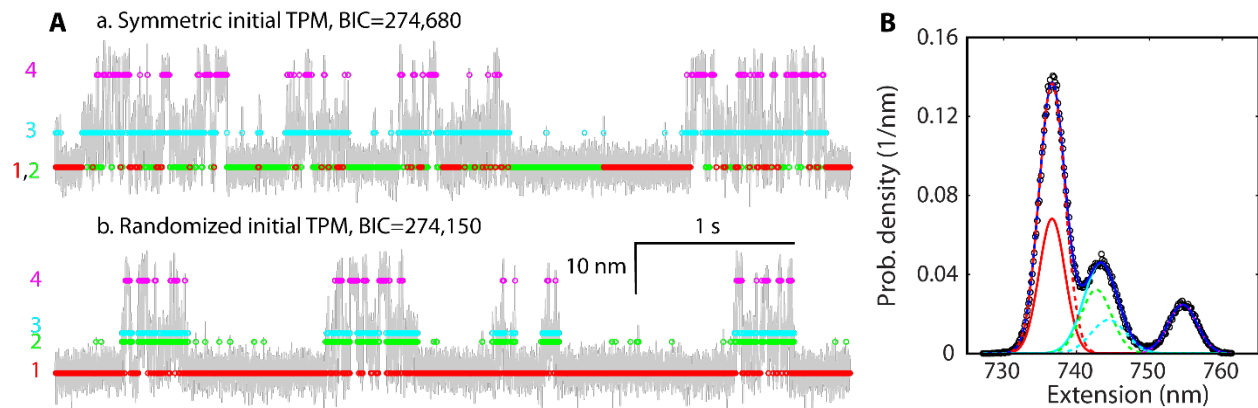

FIGURE S4   Two converged model fitting obtained by HMM-EM from different initial transition probability matrices, one is symmetric and the other is a randomized matrix.
(A) The symmetric transition probability matrix has 0.98 and 0.00667 for all diagonal and off-diagonal elements, respectively. The randomized matrix contains elements randomly generated between 0 and 1 and then normalized to one for each row. For both HMM-EM fitting, initial fitting parameters for the extensions of states 1 and 2 were the same. The symmetric TPM led to identical fitting parameters for states 1 and 2, including their extensions and transition rates. The randomized matrix led to distinct states 1 and 2.
(B) PDFs of the measured extension (circles) and their best fits (curves) by the two derived models illustrated in A. Both models fit the overall PDF well, but differed in the PDFs for individual states 1-3. The PDFs for the symmetric and randomized initial matrices are shown in solid and dashed lines, respectively.

Supporting References

1.   Rabiner, L. R. 1989. A tutorial on hidden Markov-models and selected applications in speech recognition. Proc. IEEE. 77:257-286.
4.   Qin, F., A. Auerbach, and F. Sachs. 2000. A direct optimization approach to hidden Markov modeling for single channel kinetics. Biophys. J. 79:1915-1927.
5.   Qin, F. 2014. Principles of single-channel kinetic analysis. Methods Mol Biol 1183:371-399.

9.   Syed, S., F. E. Mullner, P. R. Selvin, and F. J. Sigworth. 2010. Improved hidden Markov models for molecular motors. Part 2: Extensions and application to experimental data. Biophys. J. 2010:3696-3703.